TABLE OF CONTENTS

Section	Page No.	
1. SUMMARY	1	1/A7
2. INTRODUCTION	2	1/A8
3. LIST OF SYMBOLS	5	1/A11
4. GENERAL USER'S MANUAL	8	1/A14
4.1 Description of Program	8	1/A14
4.1.1 Differential equations and transonic scaling	8	1/A14
4.1.2 Geometry	9	1/B1
4.1.3 Boundary conditions	9	1/B1
4.1.4 Aerodynamic variables	13	1/B5
4.1.5 Numerical formulation	13	1/B5
4.1.6 Operating procedure	17	1/B9
4.1.7 Overall program flow chart	19	1/B11
4.2 Program Input	20	1/B12
4.2.1 Dictionary of input variables	20	1/B12
4.2.2 Control options	22	1/B14
4.2.3 Default values	30	1/C8
4.2.4 Input data format	31	1/C9
4.3 Error and Information Messages	34	1/C12
4.4 Program Output	38	1/D2
4.4.1 General description of the output	38	1/D2
4.4.2 Sample cases	41	1/D5
4.5 Comparisons with Data and Other Methods	42	1/D6
4.6 Program Run Times	44	1/D8
4.7 Machine Compatability	46	1/D10
5. PROGRAMMER'S MANUAL	102	2/A11
5.1 Dictionary of Subroutine Variables in Common	102	2/A11
5.2 Description of Individual Subroutines	116	2/B11
5.2.1 Subroutine ANGLE	116	2/B11
⋮ ⋮		
5.2.53 Subroutine VROOTS	145	2/E1
APPENDIX - LISTING OF COMPUTER PROGRAM TSFOIL	146	2/E2
REFERENCES	177	3/B11

DEC 1[illegible]

Stam 230-A-14

NAS 1.26:3064

COMPLETED
ORIGINAL

NASA Contractor Report 3064

Operational Manual for Two-Dimensional Transonic Code TSFOIL

Stephen S. Stahara

CONTRACT NAS2-8648
DECEMBER 1978

186

Item 830-H-14 NAS 1.24:3064

NASA Contractor Report 3064

Operational Manual for Two-Dimensional Transonic Code TSFOIL

Stephen S. Stahara
Nielsen Engineering & Research, Inc.
Mountain View, California

Prepared for
Ames Research Center
under Contract NAS2-8648

NASA
National Aeronautics
and Space Administration

Scientific and Technical
Information Office

1978

Blank Page

LIST OF ILLUSTRATIONS

Figure		Page No.
4-1	Coordinate system and airfoil geometry.	47
4-2	Summary of boundary conditions for free air flows.	48
4-3	Summary of boundary conditions for wind-tunnel simulations.	49
4-4	Sample test case 1 - program default option - subonic, free air simulation (full output): Korn airfoil, $\delta = 0.115$, $\alpha = 0.15^\circ$, $M_\infty = 0.75$.	50
4-5	Sample test case 2 - subsonic, solid wall wind-tunnel simulation (full output): NACA 0006 airfoil, $\delta = 0.06$, $\alpha = 2^\circ$, $M_\infty = 0.75$, $H = 2$.	61
4-6	Sample test case 3 - subsonic, slotted tunnel wall simulation (abbreviated output): NACA 0006 airfoil, $\delta = 0.06$, $\alpha = 2^\circ$, $M_\infty = 0.75$, $H = 2.$, $F = 0.066$.	73
4-7	Sample test case 4 - supersonic, free air simulation (abbreviated output): NACA 0006 airfoil, $\delta = 0.06$, $\alpha = 2^\circ$, $M_\infty = 1.1$.	78
4-8	Sample test case 5 - supersonic, free jet simulation (abbreviated output): NACA 0006 airfoil, $\delta = 0.06$, $\alpha = 2^\circ$, $M_\infty = 1.1$, $H = 2$.	84
4-9	Sample test case 6 - supersonic, perforated/porous wall simulation (abbreviated output): NACA 0006 airfoil, $\delta = 0.06$, $\alpha = 2^\circ$, $M_\infty = 1.1$, $H = 2$.	90
4-10	Comparison of hodograph and relaxation solutions for the surface pressure distribution on a NLR quasi-elliptical airfoil with $\delta = 0.1212$, $\alpha = 1.32^\circ$ and $M_\infty = 0.7557$.	97
4-11	Comparison of hodograph and relaxation solutions for the surface pressure distribution on a Korn airfoil with $\delta = 0.115$, $\alpha = 0.12^\circ$, and $M_\infty = 0.75$.	97
4-12	Comparison of relaxation solutions for the surface pressure distribution of a NACA 0012 airfoil at $\alpha = 2^\circ$, $M_\infty = 0.630$.	98
4-13	Comparison of conservative time-dependent and nonconservative relaxation solutions of exact isentropic equation and the small-disturbance conservative solution for the surface pressure distribution on an NACA 64A410 airfoil at $M_\infty = 0.735$ and $\alpha = 1^\circ$.	98
4-14	Comparison of FCR and NCR solutions for the similarity surface pressure distribution on a parabolic arc airfoil with $\delta = 0.06$ and $M_\infty = 0.872$. (19)	99
4-15	Comparison of various solutions for the detached bow shock wave location for a parabolic arc airfoil with $\delta = 0.06$ and $M_\infty = 1.15$. (19)	99

LIST OF ILLUSTRATIONS (CONCLUDED)

Figure		Page No.
4-16	Comparison of small disturbance FCR solutions with data for the surface pressure distribution on a Garabedian-Korn airfoil at $\alpha = 1.38^\circ$, $M_\infty = 0.768$. (19)	100
4-17	Comparison of small disturbance FCR solutions with data for the surface pressure on a NACA 0012 at $\alpha = 0^\circ$ and $M = 0.8$. (19)	101
4-18	Comparison of free-air and wind-tunnel small disturbance FCR solutions for the surface pressure distribution on a NACA 64A410 airfoil at $\alpha = 2.0^\circ$ and $M_\infty = 0.802$.	101

LIST OF TABLES

Table		Page No.
4-I	Exponents for transonic similarity parameters	47

OPERATIONAL MANUAL FOR TWO-DIMENSIONAL TRANSONIC CODE TSFOIL

By Stephen S. Stahara
Nielsen Engineering & Research, Inc.

1. SUMMARY

A detailed operating manual is presented for the computer code TSFOIL recently developed by Murman, Bailey, and Johnson at NASA/Ames Research Center. This code solves the two-dimensional, transonic, small-disturbance equations for flow past lifting airfoils in both free air and various wind-tunnel environments by using a variant of the finite-difference method initially proposed by Murman and Cole. A description of the theoretical and numerical basis of the code is provided, together with complete operating instructions and sample cases for the general user. In addition, a programmer's manual is also presented to assist the user interested in modifying the code. Included in the programmer's manual are a dictionary of subroutine variables in common and a detailed description of each subroutine.

2. INTRODUCTION

During the past several years, the rapid growth and development of numerical techniques capable of treating the nonlinear equations of transonic flow past aerodynamic configurations has generally resulted in a wealth of research computer programs but a lack of adequately documented user-oriented codes. In light of this, the present work was undertaken with the dual objectives of:

(1) providing a computer code which is capable of solving a general class of two-dimensional, transonic, small-disturbance flows, and which incorporates as many of the recent advances in the theory as possible and

(2) providing a systematically organized, user-oriented code with complete documentation to allow easy extension and modification for new applications and improvements.

A summary of the features and capabilities of the computer program was given in reference 1. This document provides an operational manual of the code, and includes a full description of the method of solution.

Although the basic solution technique used in the program is based on the original work of Krupp and Murman (refs. 2 and 3) various improvements and extensions have been incorporated in the version presented here. These include a simplified procedure for converging to the proper circulation for lifting airfoils (refs. 4 and 5), the appropriate difference operator to be used at shock waves (ref. 6), the evaluation of inviscid drag using momentum contour methods (ref. 7), wind-tunnel wall simulations (ref. 8), calculation of $M_\infty > 1$ transonic flows (ref. 9), the accelerated spread of circulation throughout the flow field for $M_\infty < 1$ free air flows (ref. 10), an analytic mesh (ref. 11) as well as a mesh cutting option to speed convergence, a successive line overrelaxation procedure based on the correction potential (ref. 12), and a pseudo-time term (ref. 12) to aid stability.

The operating manual presented here has been subdivided into two general categories. The first category is presented in section 4 and is a General User's Manual. In that section, a description of the problem solved by the code is given, including a discussion of the differential equation, geometry, boundary conditions, and numerical formulation of the problem, together with a description of the general operating procedure of the code, overall program flow chart, input/output description, error messages, and sample calculations. All the information necessary to run the code are provided in that section. In section 5, a Programmer's Manual is presented to assist the user who wishes to modify the code to incorporate additional features or to extend the code beyond its present capabilities. Included in that section is a dictionary of all subroutine variables which appear in common blocks and a detailed description of the individual subroutines. In order to provide a user-code that will readily accommodate modifications, the program has been modularized in the sense that numerous (53) special-purpose subroutines have been used, rather than one large main program. Consequently, should the user not require some of the special calculations or output provided by the code, deletions can be made simply and directly.

In reference 1, it was noted that a number of computational problems, which are repeated here for convenience, existed in the version of the code then available. Some of those problems have since been alleviated by improvements to that initial version. In particular, the computation of lifting airfoils in supersonic free streams, which had proved troublesome, has been improved by the addition of a pseudo-time term as discussed in detail in section 4.1.5, and also by the introduction of an analytical mesh suggested in reference 11. A characteristic difficulty with $M_\infty > 1$ flows is that the maximum iteration error may initially decrease monotonically to a certain level, and then oscillate about that level, or in more severe cases, diverge. The maximum error almost always occurs at points near the airfoil and close to the bow or trailing edge shocks, or the locations where those shocks cross the mesh outer boundary. Addition of the pseudo-time term has served to stabilize this feature of the calculation, particularly when mesh cutting has been employed and the switch is made to the next finer mesh. Use of the analytic mesh frequently has served to preserve the monotonic convergence behavior of the solution, presumably both by the clustering of points near the leading and trailing edges and the expansion of points near the mesh boundaries.

Addition of these two features has also served to enhance the convergence of $M_\infty < 1$ flows. For example, converged solutions were achieved for a number of transonic flows past the 64A410 airfoil which could not previously be obtained with the earlier version. Also, supercritical flows for which the shock wave on the airfoil upper surface reached the trailing edge had proven difficult to calculate. For these cases, use of the analytic mesh appears to aid convergence considerably.

However, the occurrence of irregularities in the solution near the leading edge, as noted in reference 1, remains in the present version. Although these also appear in Krupp's results (ref. 2), in the current program they are predicted to be somewhat stronger. These dissimilarities may be due to the slight differences in numerical procedures used. Of course, one of the inherent deficiencies in any first-order, small-disturbance theory is the inaccurate treatment of the leading edge. A more accurate treatment, such as those suggested in references 13 and 14, has not been incorporated in the program, although a Reigel's rule option has been added.

The program has been written in FORTRAN IV source language and developed for use on a CDC 7600 computer using the OPT=2 compiler. The memory requirements are $(57K)_8$ and $(30K)_8$ words for small core and large core, respectively.

3. LIST OF SYMBOLS

$A_{j,i}$	coefficient matrix, equation (36)
c	airfoil chord
$\bar{C}_i$	column vector containing values of the correction potential $C_{j,i}$ along the i^{th} column of the finite-difference mesh, equation (47)
C_D	drag coefficient, equation (34)
$C_{j,i}$	correction potential, equation (42)
C_m	pitching-moment coefficient, equation (33)
C_p	pressure coefficient, equation (28)
C_{p_u}	pressure coefficient associated with the airfoil upper surface
C_{p_ℓ}	pressure coefficient associated with the airfoil lower surface
f_i	column vector associated with the i^{th} column of the finite-difference grid, equation (46)
F	nondimensional wind-tunnel slot parameter, equation (20)
$F_{u,\ell}$	dimensionless affine function describing, respectively, the ordinates of the upper and lower airfoil surface, equation (7)
H	ratio of wind-tunnel half height to airfoil chord
$\tilde{H}$	transonically scaled tunnel half height to airfoil chord ratio, equation (22)
i,j	indices identifying the grid point at the i^{th} column and j^{th} row
k	exponent of Mach number in similarity definition of the small-disturbance equation, equations (2) and (6)
K	transonic similarity parameter, equation (5)
L	nondimensional spacing of slot centerlines, normalized by c; equation (20)
m	exponent of Mach number in similarity definition of transonically scaled y coordinate, equation (4)
M	local Mach number, equation (30)
M_∞	free-stream Mach number
n	exponent of Mach number in similarity definition of velocity potential, equation (1)

p	pressure
P	nondimensional experimentally determined wind-tunnel porosity parameter equal to $2\Phi_y(x,\pm H)/\Delta p_{wall}(x,\pm H)/q_\infty$, equation (19)
$\tilde{P}$	transonically scaled wind-tunnel porosity parameter, equation (23)
P_c	shock-wave total pressure loss
P_∞	free-stream pressure
q_∞	free-stream dynamic pressure
$R_{j,i}$	residual matrix, equation (45)
S	nondimensional slot width, normalized by c; equation (20)
U_∞	free-stream velocity
(x,y)	nondimensional Cartesian coordinate system with x axis directed rearward and aligned with the free stream; coordinates normalized by c
(x',y')	equal to (cx,cy)
$\tilde{y}$	transonically scaled y coordinate, equation (4)
$y_{u,l}$	ordinates of the upper (u) and lower (l) airfoil surface, equation (7)
α	angle of attack
Γ	circulation, equation (12)
Γ_{ff}	far-field circulation, equation (13)
γ	ratio of specific heats
δ	airfoil thickness ratio
ϵ	parameter multiplying pseudo-time term $\Delta t\phi_{xt}/\Delta x$ in small-disturbance partial differential equation, equation (48)
θ	local flow angle, equation (29)
$\mu_{j,i}$	switching parameter, equation (37)
Φ	disturbance velocity potential, equation (1)
$\bar{\Phi}$	total velocity potential, equation (1)
ϕ	dimensionless perturbation velocity potential, equation (1)
ϕ_{ff}	asymptotic far-field solution of the perturbation velocity potential

$\Delta\phi(x)$	jump in perturbation velocity potential along trailing vortex wake, equations (14) and (16)
$\Delta\phi_{te}$	jump in perturbation velocity potential at the trailing edge, equation (12)
ω	relaxation parameter, equation (44)
ω_e	relaxation parameter for elliptic points, equation (44)
ω_Γ	relaxation parameter for circulation, equation (13)

4. GENERAL USER'S MANUAL

4.1 Description of Program

This section provides an overall description of the program, a brief discussion of the theoretical and numerical formulation of the problem solved by the code, including a description of the differential equation, airfoil geometry, boundary conditions, and finite-difference method. Also, the general operating procedure of the code is reviewed and an overall program flow chart is provided.

4.1.1 Differential equations and transonic scaling.- The partial differential equation solved by TSFOIL is the transonic small-disturbance equation for the perturbation velocity potential cast in terms of transonic similarity variables. Because of the nonuniqueness of the scaling, various forms have been adopted by different authors. If (Φ,ϕ) are, respectively, the physically and transonically scaled perturbation potentials, the various forms can be combined in the following general manner:

$$\bar{\Phi}(x',y') = cU_\infty[x + \Phi(x,y) + \ldots] = cU_\infty\left[x + \frac{\delta^{2/3}}{M_\infty^n}\phi(x,\tilde{y}) + \ldots\right] \tag{1}$$

where c is the airfoil chord, U_∞ and M_∞, respectively, are the free-stream velocity and Mach number, δ is the airfoil thickness ratio, (x,y) are the physical Cartesian coordinates (x',y') normalized by airfoil chord, and $\tilde{y}$ the transonically scaled y coordinate given below. Now, (Φ,ϕ) satisfy

$$\left[(1 - M_\infty^2)\Phi_x - \frac{\gamma + 1}{2} M_\infty^k \Phi_x^2\right]_x + \left[\Phi_y\right]_y = 0 \tag{2}$$

$$\left[K\phi_x - \frac{\gamma + 1}{2} \phi_x^2\right]_x + \left[\phi_{\tilde{y}}\right]_{\tilde{y}} = 0 \tag{3}$$

where

$$\tilde{y} = \delta^{1/3} M_\infty^m y \tag{4}$$

$$K = \frac{1 - M_\infty^2}{\delta^{2/3} M_\infty^{2m}} \tag{5}$$

and the exponents (k,m,n) are related by

$$k = 2m + n \tag{6}$$

The different forms of the transonic similarity variables depend essentially on the particular choice of the exponents (k,m,n), and Table I provides a summary of the particular values proposed by Cole, Spreiter, or Krupp which have been incorporated into the program. The partial differential equation actually solved by TSFOIL is then equation (3).

4.1.2 Geometry.- The coordinate system used in the program is the Cartesian system shown in figure 4-1, with the airfoil located along the x-axis between 0 and 1, at angle of attack, α, and with a shape given by

$$y_{u,\ell}(x) = \delta F_{u,\ell}(x) \qquad 0 \leq x \leq 1 \tag{7}$$

where (u,ℓ) denotes the upper and lower surfaces, and $F_{u,\ell}(x)$ is a function of order one describing the ordinates of a family of affinely related airfoils. Default profiles incorporated into the program are the parabolic-arc

$$F_{u,\ell}(x) = \pm 2x(1-x) \tag{8}$$

the NACA four-digit series

$$F_{u,\ell}(x) = \pm(1.4845\sqrt{x} - 0.63\,x - 1.758\,x^2 + 1.415\,x^3 - 0.5075\,x^4) \tag{9}$$

and the Korn airfoil described in reference 15. Provision has been made in the program for the user to input a profile of his choice by either a formula or a table of ordinates.

4.1.3 Boundary conditions.- The boundary conditions applied to this problem consist of the flow tangency condition at the airfoil surface, a condition for lifting airfoils assuring the value of circulation be unique, and some far-field conditions to be satisfied on the outer boundaries of the computational mesh. Within the framework of small-disturbance theory, the tangency condition at the airfoil can be linearized and applied along the slit $\tilde{y} = 0$, $0 \leq x \leq 1$ as

$$\phi_{\tilde{y}}(x,0^+) = \frac{dF_u(x)}{dx} - \frac{\alpha}{\delta} \tag{10}$$

$$\phi_{\tilde{y}}(x,0^-) = \frac{dF_\ell(x)}{dx} - \frac{\alpha}{\delta} \tag{11}$$

For lifting airfoils, two options have been incorporated in the program to insure that the circulation is unique. The first is the usual Kutta condition which is satisfied by requiring that ϕ_x (pressure) be continuous along the line $\tilde{y} = 0$, $x \geq 1$, and that $\phi_{\tilde{y}}$ (flow angle) be continuous across $\tilde{y} = 0$, $x > 1$. The disturbance potential ϕ remains single-valued by introduction of a cut along the $\tilde{y} = 0$ axis downstream of the airfoil along which ϕ is discontinuous, jumping by the value of the circulation Γ defined by

$$\Gamma = -\oint d\phi = \Delta\phi_{te} \tag{12}$$

where the contour is taken around the airfoil surface, and the resultant integral is equal to the jump in potential at the trailing edge. The circulation Γ_{ff} used in the asymptotic far-field solutions to update the outer boundaries is determined from the relation

$$\Gamma_{ff}^{n+1} = \Gamma_{ff}^{n} - \omega_\Gamma \left(\Gamma_{ff}^{n} - \Delta\phi_{te}^{n}\right) \tag{13}$$

where ω_Γ is the relaxation factor for circulation and n is the iteration count. The updated jump in potential $\Delta\phi(x)$ along the slit $\tilde{y} = 0$, $x > 1$ is calculated by the program according to the formula

$$\Delta\phi(x) = \Delta\phi_{te}^{n} + \frac{x - 1}{x_{max} - 1}\left(\Gamma_{ff}^{n+1} - \Gamma_{ff}^{n}\right) \tag{14}$$

From the results of reference 4, a value of circulation relaxation factor $\omega_\Gamma = 1.0$ (default value) is recommended. This implies that the updated far-field circulation be set equal to the new jump in potential at the trailing edge,

$$\Gamma_{ff}^{n+1} = \Delta\phi_{te}^{n} \tag{15}$$

and that the jump in potential along the trailing vortex wake be held constant and equal to

$$\Delta\phi(x) = \Delta\phi_{te}^{n} \tag{16}$$

Numerical experimentation has shown that the simpler representation of equations (15) and (16) gives the same results and convergence rate as both the method of Krupp (ref. 2) and the closely related procedure of equations (13) and (14).

The second option available to insure uniqueness of circulation is to specify the lift. In this case, the Kutta condition is not satisfied, and the pressure becomes double-valued at the trailing edge. The jump in potential at the trailing edge, as well as the far-field circulation, and the potential jump along the trailing vortex wake are given by

$$\Delta\phi_{te} = \Gamma_{ff} = \Delta\phi(x) = \frac{(C_L)_{SET}}{2} \tag{17}$$

where $(C_L)_{SET}$ is the specified lift coefficient.

For conditions at the outer boundaries, provision is made in the program to represent free air flows and various wind-tunnel wall simulations for both $M_\infty < 1$ and $M_\infty > 1$. For subsonic free air flows, the program incorporates the asymptotic far-field solution ϕ_{ff} given by a compressible vortex and doublet, while for supersonic free air flows the appropriate far-field conditions incorporated are that the perturbation velocities $\phi_x, \phi_{\tilde{y}}$ vanish at the upstream boundary and that the outgoing wave condition

$$\phi_{\tilde{y}} = \pm\sqrt{K}\,\phi_x \tag{18}$$

be enforced at the top and bottom boundaries in order to prevent reflected waves. Strictly speaking, for $M_\infty > 1$, no condition is required at the downstream boundary if as we assume it is sufficiently far removed downstream so that the outflow is entirely supersonic. However, for actual numerical application, some condition is needed (ref. 8) to treat subsonic outflow which may develop during the course of the relaxation solution process prior to convergence. By numerical experimentation, the condition $\phi_x = 0$ was found to be satisfactory. Figure 4-2 summarizes these conditions for free air calculations.

For wind-tunnel simulations, the classical homogeneous tunnel wall boundary condition

$$\Phi_x(x,\pm H) \pm FH\Phi_{xy}(x,\pm H) \pm \frac{1}{P}\,\Phi_y(x,\pm H) = 0 \tag{19}$$

is applied at the tunnel walls $y = \pm H$. Here, H is the ratio of tunnel half height to airfoil chord, F is the tunnel slot parameter, and P is the porosity parameter which accounts for viscous effects in the slot and must be determined experimentally. A simplified analysis has been made by Baldwin, et al. (ref. 16) for the slot parameter F, which he gives as

$$F = \frac{L}{\pi H} \ell n \left[\csc \left(\frac{\pi S}{2L} \right) \right] \qquad (20)$$

where L is the spacing between slot centerlines and S is the slot width, both normalized by airfoil chord. The program uses equation (19) in the similarity form

$$\phi_x(x,\pm\tilde{H}) \pm F\tilde{H}\phi_{x\tilde{y}}(x,\pm\tilde{H}) \pm \frac{1}{\tilde{P}} \phi_{\tilde{y}}(x,\pm\tilde{H}) = 0 \qquad (21)$$

where $(\tilde{H},\tilde{P})$ are the transonically scaled tunnel half height and porosity given by

$$\tilde{H} = \delta^{1/3} M_\infty^m H \qquad (22)$$

$$\tilde{P} = P/(\delta^{1/3} M_\infty^m) \qquad (23)$$

The following subcases of equation (19) have been incorporated into the program as separate options

$\phi_{\tilde{y}}(x,\pm\tilde{H}) = 0$	Solid Wall	(24)
$\phi_x(x,\pm\tilde{H}) = 0$	Free Jet	(25)
$\tilde{P}\phi_x(x,\pm\tilde{H}) \pm \phi_{\tilde{y}}(x,\pm\tilde{H}) = 0$	Ideal Perforated/ Porous Wall	(26)
$\phi_x(x,\pm\tilde{H}) \pm F\tilde{H}\phi_{x\tilde{y}}(x,\pm\tilde{H}) = 0$	Ideal Slotted Wall	(27)

while the general case has not yet been included. For tunnel flows, the upstream and downstream conditions used are the following. For $M_\infty < 1$, the asymptotic far-field solution ϕ_{ff} which satisfies equation (19) and the linearized version of the differential equation (3) is used. The case of choked flow, as treated in reference 17, has not been included in the program. For $M_\infty > 1$, the conditions $\phi_x, \phi_{\tilde{y}} = 0$ are enforced at the upstream boundary, while the flow is required to be supersonic at the downstream boundary and the condition $\phi_x = 0$ is employed, as in the supersonic free air calculation. These conditions are summarized in figure 4-3.

4.1.4 Aerodynamic variables.- The aerodynamic quantities of interest output by the program are given below in terms of the small-disturbance similarity potential $\phi(x,\tilde{y})$. These include pressure coefficient, C_P, flow angle, θ, local Mach number, M, shock wave total pressure loss, $[P_O]/P_\infty$, and coefficients of lift, C_L, pitching moment, C_m, and drag, C_D.

$$C_p = \delta^{2/3} M_\infty^{-n} (-2\phi_x) \tag{28}$$

$$\theta = \delta\phi_{\tilde{y}} \tag{29}$$

$$M = \left\{1 - \left[K - (\gamma + 1)\phi_x\right]\delta^{2/3} M_\infty^{2m}\right\}^{1/2} \tag{30}$$

$$\frac{[P_O]}{P_\infty} = 1 + \delta^2 \frac{\gamma(\gamma + 1)}{12} [\phi_x]_s^3 \tag{31}$$

$$C_L = -\int_0^1 (C_{PU} - C_{P\ell})\,dx \tag{32}$$

$$C_m = \int_0^1 (x - x_m)(C_{Pu} - C_{P\ell})\,dx \qquad \text{(positive nose-up)} \tag{33}$$

$$C_D = \delta^{5/3} M_\infty^{-n} \left\{ \oint_C \left[K\phi_x^2 - \phi_{\tilde{y}}^2 - \frac{2}{3}(\gamma + 1)\phi_x^3\right] d\tilde{y} - \phi_x\phi_{\tilde{y}}\,dx - \frac{\gamma + 1}{6} \int_{S\supset C} [\phi_x]_s^3\, d\tilde{y}\right\} \tag{34}$$

where $[\phi_x]_s$ is the jump in the streamwise perturbation velocity across a shock.

4.1.5 Numerical formulation.- The numerical formulation of the problem together with the techniques employed to solve it are presented in this section. The numerical solution of the problem is initiated by replacing equation (3) by the following finite-difference form given by Bailey and Ballhaus in reference 18.

$$\Big\{(1-\mu_{j,i})\, A_{j,i}\Big[(\phi_x)_{j,i+1/2} - (\phi_x)_{j,i-1/2}\Big]$$

$$+ \mu_{j,i-1}\, A_{j,i-1}\Big[(\phi_x)_{j,i-1/2} - (\phi_x)_{j,i-3/2}\Big]\Big\} \Big/ (x_{i+1/2} - x_{i-1/2})$$

$$+ \Big[(\phi_y)_{j+1/2,i} - (\phi_y)_{j-1/2,i}\Big] \Big/ (y_{j+1/2} - y_{j-1/2}) = 0 \qquad (35)$$

where

$$A_{j,i} = 1 - M_\infty^2 - (\gamma + 1)M_\infty^k(\phi_x)_{j,i} \qquad (36)$$

$$\mu_{j,i} = \begin{cases} 0 & \text{for} \quad A_{j,i} > 0 \\ 1 & \text{for} \quad A_{j,i} < 0 \end{cases} \qquad (37)$$

$$(\phi_x)_{j,i+1/2} = \frac{\phi_{j,i+1} - \phi_{j,i}}{x_{i+1} - x_i} \qquad (38)$$

$$(\phi_y)_{j+1/2,i} = \frac{\phi_{j+1,i} - \phi_{j,i}}{y_{j+1} - y_j}\text{, etc.} \qquad (39)$$

and $(\phi_x)_{j,i}$ in equation (36) is approximated by

$$(\phi_x)_{j,i} = \frac{1}{2}\Big[(\phi_x)_{j,i+1/2} + (\phi_x)_{j,i-1/2}\Big] \qquad (40)$$

This form of the difference equation with the switching parameter $\mu_{j,i}$ encompasses the four operators of Murman's fully conservative relaxation (FCR) method (ref. 15); that is, elliptic ($\mu_{j,i-1} = 0$, $\mu_{j,i} = 0$), hyperbolic ($\mu_{j,i-1} = 1$, $\mu_{j,i} = 1$), parabolic ($\mu_{j,i-1} = 0$, $\mu_{j,i} = 1$), and shock point ($\mu_{j,i-1} = 1$, $\mu_{j,i} = 0$), and for evenly spaced meshes is identical to that method. For unequally spaced meshes, the first term differs from Murman's (ref. 19) by a factor of $(x_i - x_{i-2})/(x_{i+1} - x_{i-1})$ in hyperbolic regions. The finite-difference equation (35) can be cast in the following form

$$(1-\mu_{j,i})A_{j,i}(CXXL_i\phi_{j,i-1} - CXXC_i\phi_{j,i} + CXXR_i\phi_{j,i+1})$$

$$+ \mu_{j,i-1}A_{j,i-1}(CXXL_{i-1}\phi_{j,i-2} - CXXC_{i-1}\phi_{j,i-1} + CXXR_{i-1}\phi_{j,i})$$

$$+ CYYD_j\phi_{j-1,i} - CYYC_j\phi_{j,i} + CYYU_j\phi_{j+1,i} = 0 \qquad (41)$$

where the coefficients ($CXXL_i$, $CXXC_i$, etc.) depend on mesh geometry only and can be deduced from equation (35). Although equation (41) forms the basis of the numerical analysis, the solution technique actually employed in the program is similar to that recently suggested by Jameson (ref. 12) in which the difference equation (41) is recast in terms of a correction potential $C_{j,i}$ defined by

$$\phi^{n+1}_{j,i} = \phi^{n}_{j,i} + C_{j,i} \qquad (42)$$

where n represents the current iteration cycle.

The governing equation for $C_{j,i}$ is determined in the following fashion. First, we consider the difference equation (41) in which the values $\phi_{j,i}$ of the potential along the current column are replaced by their provisional values $\tilde{\phi}_{j,i}$, while the values ($\phi^{n+1}_{j,i-1}$, $\phi^{n+1}_{j,i-2}$, $\phi^{n}_{j,i+1}$, $A^{n}_{j,i-1}$, $A^{n}_{j,i}$) are used to evaluate the remaining terms. Similarly, we define a residual $R_{j,i}$ by considering the same equation but with the result $\phi^{n}_{j,i}$ used for $\phi_{j,i}$. By subtracting those two equations and then eliminating $\tilde{\phi}_{j,i}$ by use of the overrelaxation formula

$$\phi^{n+1}_{j,i} = \phi^{n}_{j,i} + \omega\,(\tilde{\phi}_{j,i} - \phi^{n}_{j,i}) \qquad (43)$$

where

$$\omega = \left\{ \begin{array}{lll} \omega_e,\ 1 < \omega_e < 2 & \text{for locally subsonic flow} & (A_{j,i} > 0) \\ & & \\ 1 & \text{for locally supersonic flow} & (A_{j,i} < 0) \end{array} \right\} \qquad (44)$$

the following difference equation results for $C_{j,i}$:

$$\left\{-(1-\mu_{j,i})A^{n}_{j,i}CXXC_i/\omega_e + \mu_{j,i-1}A^{n}_{j,i-1}CXXR_{i-1}\right\}C_{j,i}$$

$$+ CYYD_j C_{j-1,i} + CYYU_j\, C_{j+1,i} = -R_{j,i} \qquad (45)$$

We note that in the derivation of equation (45), new values $\phi^{n+1}_{j,i}$ rather than provisional values $\tilde{\phi}_{j,i}$ are used to evaluate ϕ_{yy} at both subsonic and supersonic points. As pointed out in reference 12, this avoids a discontinuity in ϕ_{yy} at a sonic line.

The solution of equation (45) then proceeds by using an iterative successive line over-relaxation (SLOR) algorithm as follows. The system of equations for each vertical line is written as

$$A\bar{C}_i = f_i \tag{46}$$

where $\bar{C}_i$ is the J dimensional column vector

$$\bar{C}_i = \begin{pmatrix} C_{1,i} \\ \cdot \\ \cdot \\ \cdot \\ C_{j,i} \end{pmatrix} \tag{47}$$

A is a J × J dimensional, diagonally dominant, tridiagonal matrix, and f_i is a J dimensional column vector. The solution process consists of making an initial guess for ϕ, and then successively sweeping the grid from the upstream to downstream boundary, solving for new values of $C_{j,i}$ from equation (46) by direct elimination using the method of triangular decomposition, and then determining new values of ϕ from equation (42). The initial guess for ϕ may be selected from a uniform flow $(\phi = 0)$ or a previous case stored in core or on a peripheral unit. With regard to the over-relaxation parameter ω_e, acceptable values on a fine mesh for ω_e are $1.8 < \omega_e < 1.95$, although optimum values for particular applications must be determined by numerical experimentation.

A mesh refinement option has been incorporated into the program to enhance the rate of convergence. Under this option, the user may elect to initiate the calculation either on a mesh (medium mesh) obtained by deleting every other mesh point of the input mesh (fine mesh), or on an even coarser mesh (coarse mesh) obtained by again deleting every other point on the medium mesh. In the latter case, the iterative solution is started on the coarse mesh, and after appropriate convergence is reached, continued on the medium mesh, and then completed on the input mesh. Also, in order to improve stability and aid convergence, an option has been provided in the program to add the pseudo-time term in the form

$$- \epsilon \frac{\Delta t}{\Delta x} \phi_{xt} \tag{48}$$

where

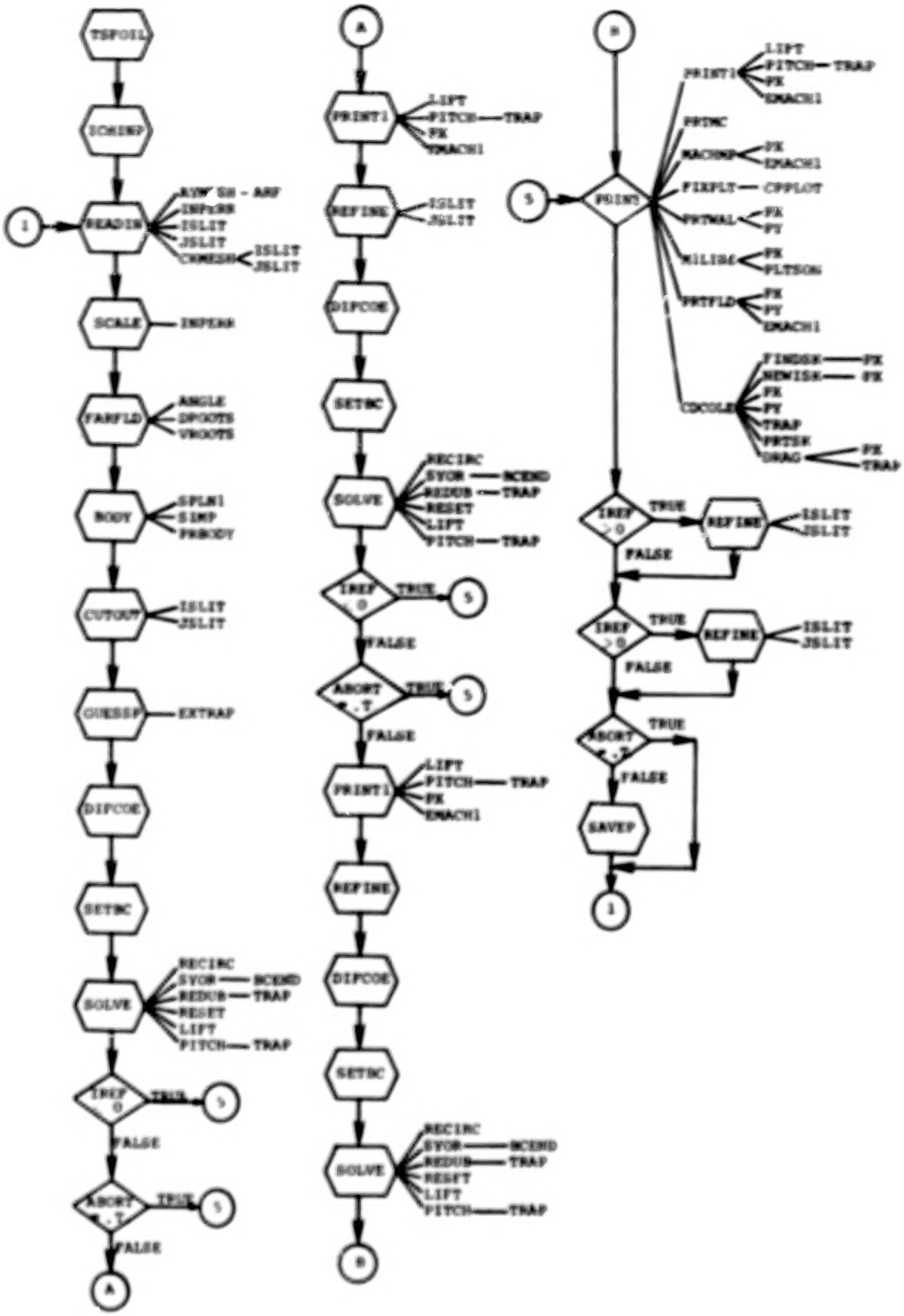
$$\frac{\Delta t}{\Delta x}\phi_{xt} = \frac{\left(\phi^{n+1}_{j,i} - \phi^{n}_{j,i}\right) - \left(\phi^{n+1}_{j,i-1} - \phi^{n}_{j,i-1}\right)}{\left(x_i - x_{i-1}\right)^2} = \frac{C_{j,i} - C_{j,i-1}}{\left(x_i - x_{i-1}\right)^2} \qquad (49)$$

to the left-hand side of equation (45) as suggested in references 12 and 18. The parameter ϵ is an input parameter with a value in the range $0 \leq \epsilon \leq 1$. Considerable improvement in stability and convergence has been obtained in a number of both subsonic and supersonic free-stream calculations when this term is added. However, this result is not true for all cases and extensive numerical experimentation to determine optimum values for ϵ has not yet been carried out. Consequently, the user must, of necessity, determine suitable values for his particular application. An appropriate first guess would be $\epsilon = 0.2$ which is the default value.

Finally, we note that the numerical methods used in this program are of the shock-capturing, rather than shock-fitting type. This means that shock waves are spread over several mesh intervals and do not appear as sharp discontinuities. Calculations have shown (ref. 19) that for subsonic flow downstream of shock waves, four mesh points are sufficient to capture the shock jump; whereas for supersonic downstream flow, six to ten points are required. The latter is due to dissipation effects which dominate the first-order accurate hyperbolic operator in equation (35), and smooth out shocks to that extent. This is a weakness in the method if sharp supersonic oblique shocks are required, and necessitates special indentations of the momentum contours used to calculate the inviscid drag for all cases where $M_\infty > 1$, and also for cases where $M_\infty < 1$ when tail plus wake shocks occur. See subroutine CDCOLE in section 5.2 for a description of the momentum contours incorporated in the program.

4.1.6 Operating procedure.- The general operating procedure of the program is straightforward and is described in this section. First, the main program prints all of the input cards (ECHINP), then reads the input for the first case, checks it for errors, assigns default values to those parameters not specified, and then prints all of the input parameters (READIN). Next, if the input was in physical rather than transonic form, the appropriate scaling is applied to render all the necessary variables into transonic similarity form (SCALE). The unit-strength boundary data on the far-field mesh boundaries are then computed (FARFLD). Next, the geometrical characteristics of the airfoil that are required in the finite-difference calculations are determined (BODY). The medium and coarse meshes are then established if the user has opted to use the mesh refinement option (CUTOUT). Next, the ϕ array is initialized (GUESSP), and then the various difference coefficients that depend on mesh size are calculated for the current mesh (DIFCOE). The I and J indices for solving the difference equations are set, and the airfoil slope boundary condition is multiplied by mesh spacing constants for use in the finite-difference calculation (SETBC). The relaxation solution of the difference equations is then carried out in an iterative fashion by sweeping the flow field from upstream to downstream boundaries with convergence information being output after a specified number of sweeps (SOLVE). If the calculations have been carried out on an intermediate mesh, flow field information on the dividing streamline, including the airfoil, are printed (PRINT1), and then mesh points are added to obtain the next finer mesh (REFINE). The sequence of relaxation calculations are then repeated (DIFCOE, SETBC, SOLVE, PRINT1) until the final mesh is reached. At this point, the final flow information is output (PRINT), the solution stored for later use (SAVEP), and the calculation for the next case begun. If the solution should diverge during the course of the finite-difference calculation (SOLVE), the computation is then aborted, with portions of the final printout (PRINT) provided to assist the user in determining the cause of the difficulty.

4.1.7 Overall program flow chart.- An overall program flow chart is provided below to illustrate the logical flow of the code.


TSFOIL
ICHINP
1
READIN
RYW SH - ARF
INPERR
ISLIT
JSLIT
CKMESH
ISLIT
JSLIT
SCALE
INPERR
FARFLD
ANGLE
DROOTS
VROOTS
BODY
SPLN1
SIMP
PRBODY
CUTOUT
ISLIT
JSLIT
GUESSP
EXTRAP
DIFCOE
SETBC
SOLVE
RECIRC
SYOR
BCEND
REDUB
TRAP
RESET
LIFT
PITCH
TRAP
IREF
< 0
TRUE
5
FALSE
ABORT
= .T
TRUE
5
FALSE
A
A
PRINT1
LIFT
PITCH
TRAP
FX
EMACH1
REFINE
ISLIT
JSLIT
DIFCOE
SETBC
SOLVE
RECIRC
SYOR
BCEND
REDUB
TRAP
RESET
LIFT
PITCH
TRAP
IREF
< 0
TRUE
5
FALSE
ABORT
= .T
TRUE
5
FALSE
PRINT1
LIFT
PITCH
TRAP
FX
EMACH1
REFINE
DIFCOE
SETBC
SOLVE
RECIRC
SYOR
BCEND
REDUB
TRAP
RESET
LIFT
PITCH
TRAP
B
B
5
PRINT
PRINT1
LIFT
PITCH
TRAP
FX
EMACH1
PRTMC
MACHMP
FX
EMACH1
FIXPLT
CPPLOT
PRTWAL
FX
FY
MILINE
FX
PLTSON
PRTFLD
FX
FY
EMACH1
CDCOLE
FINDSK
FX
NEWISK
FX
FX
FY
TRAP
PRTSK
DRAG
FX
TRAP
IREF
> 0
TRUE
REFINE
ISLIT
JSLIT
FALSE
IREF
> 0
TRUE
REFINE
ISLIT
JSLIT
FALSE
ABORT
= .T
TRUE
FALSE
SAVEP
1

4.2 Program Input

The input required by the program is described in this section. A dictionary of input variables is provided first, followed by a description of the various input control options available to the user. All of the default values of the input variables are given next, followed by a description of the input data format including preparation of a sample run.

4.2.1 Dictionary of input variables.- All variables that are input to this program are described in the following list.

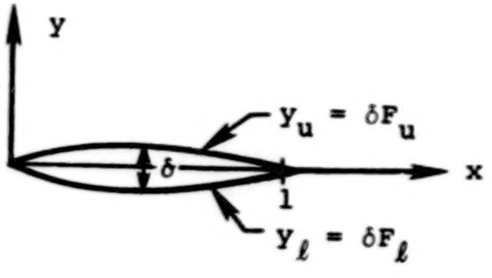
VARIABLE	DESCRIPTION
AK	transonic similarity parameter; equal to $1-M_\infty^2/M_\infty^{2m}\delta^{2/3}$ where m is defined in Table 4-I according to the type of transonic scaling used; input required only if transonic scaling used in input/output; that is, PHYS = .F.
ALPHA	angle of attack, α, in degrees for PHYS = .T.
AMESH (logical)	control option for specification of analytical default mesh; equal to .T. or .F.
BCFOIL (integer)	control option for specification of airfoil geometry; equal to 1, 2, 3, or 4
BCTYPE	control option for type of flow environment; equal to 1, 2, 3, 4, or 5
CLSET	specified lift coefficient; input only if Kutta condition not enforced; that is, KUTTA = .F.
CVERGE	convergence criterion for maximum iteration error in ϕ
DELTA	airfoil thickness ratio δ
DVERGE	divergence criterion for maximum iteration error in ϕ
EMACH	free-stream Mach number, M_∞; because of the particular far-field formulation used in this program, M_∞ can be close but not exactly equal to 1
EPS	coefficient of pseudo-time term $\Delta t\phi_{xt}/\Delta x$ in differential equation
F	slot parameter for ideal slotted wind tunnel
FCR (logical)	control option for specification of type of differencing to be used at shock waves; equal to .T. or .F.
GAM	ratio of specific heats γ
H	ratio of tunnel half height/airfoil chord
ICUT	control option for mesh cutting and refinement; equal to 0, 1, or 2

VARIABLE	DESCRIPTION
IMAXI	number of x-mesh points in input mesh; IMAXI $\leq$ 100
IMIN	value of I index designating the x-mesh point where the finite-difference calculation is to start
IPRTER	control option for print frequency of intermediate output
JMAXI	number of y-mesh points in input mesh; JMAXI $\leq$ 100
JMIN	value of J index designating the y-mesh point where the finite-difference calculation is to start
KUTTA (logical)	control for Kutta condition specification; equal to .T. or .F.
MAXIT	maximum allowed number of iteration sweeps through the flow field
NL	number of ordinate points used to describe airfoil lower surface; NL $\leq$ 100
NU	number of ordinate points used to describe airfoil upper surface; NU $\leq$ 100
PHYS (logical)	control for specification of the type of scaling (physical or transonic) used in input/output; equal to .T. or .F.
POR	porosity factor for ideal porous/perforated wind-tunnel wall
PRTFLO	control for printout of final flow field; equal to 1, 2, or 3
PSAVE (logical)	control for saving restart block of values on unit 3; equal to .T. or .F.
PSTART (integer)	control for initialization of ϕ array; equal to 1, 2, or 3
SIMDEF (integer)	control for definition of transonic similarity scaling; equal to 1, 2, 3, or 4
WCIRC	relaxation factor for circulation
WE(I)	vector array of length 3 representing values of the relaxation factor for elliptic points on the coarse, medium, and fine meshes, respectively; if specified, all three values must be given
XGRDIN	control for input of user-designated x-grid; equal to .T. or .F.

VARIABLE	DESCRIPTION
XIN(I)	vector array of length IMAXI of user-supplied x-mesh points; normalized by airfoil chord; maximum of 100
XL(I)	vector array of length NL of x-ordinates of the airfoil lower surface; normalized by airfoil chord; maximum of 100
XU(I)	vector array of length NU of x-ordinates of the airfoil upper surface; normalized by airfoil chord; maximum of 100
YGRDIN	control for input of user-designated y-grid; equal to .T. or .F.
YIN(I)	vector array of length JMAXI of user-supplied y-mesh points, normalized by airfoil chord for PHYS = .T.; maximum of 100
YL(I)	vector array of length NL of y-ordinates of the airfoil lower surface; normalized by airfoil chord times thickness ratio; maximum of 100
YU(I)	vector array of length NU of y-ordinates of the airfoil upper surface; normalized by airfoil chord times thickness ratio; maximum of 100

4.2.2 Control options.- Those input variables which allow the user to select the various options incorporated in this program are detailed in this section together with the options they control.

Airfoil geometry option: Four choices are available to the user for the specification of the airfoil geometry. Selection is made according to the value specified (1, 2, 3, or 4) for the integer index BCFOIL. For BCFOIL = 1, the program uses the symmetric NACA four-digit (00XX) series geometry; for BCFOIL = 2, a parabolic-arc profile is used; for BCFOIL = 3, user-supplied values of upper and lower airfoil ordinates (YU(I), YL(L)) versus abcissas (XU(I), XL(I)) are required by the program, which then calculates internally the ordinates and slopes at the input mesh points by cubic spline interpolation--under this geometry option, a default airfoil (Korn) is provided; for BCFOIL = 4, the user provides his own specification (i.e., ordinates and slopes at the input mesh points) of the airfoil geometry. We note that the last option (BCFOIL = 4) is, of course, open (inactive) and requires user-supplied information in subroutine BODY. If a formula is available for the airfoil profile desired, this option should be used. Implementation is straightforward and an illustrative example is given below. Consider the symmetric profile formed by the sinusoidal curve given by:



$$y_{u,\ell}(x) = \pm \frac{\delta}{2} \sin(\pi x)$$

so that

$$F_{u,\ell}(x) = \pm \frac{1}{2} \sin(\pi x)$$

$$\frac{dF_{u,\ell}(x)}{dx} = \pm \frac{\pi}{2} \cos(\pi x)$$

Thus, the following FORTRAN statement changes would be entered into subroutine BODY. Immediately after statement number 400 (see the listing of subroutine BODY in the Appendix), the abnormal stop message and STOP card would be removed and the following statements inserted:

```
          PI = 3.1415927
          IC = 0
          DO 425 I = ILE, ITE
          IC = IC + 1
          Z = XIN(I)
          Z1 = PI * Z
          XFOIL(IC) = Z
          FU(IC) = SIN(Z1)/2.
          FL(IC) = - FU(IC)
          FXU(IC) = PI * COS(Z1)/2.
          FXL(IC) = - FXU(IC)
      425 CONTINUE
```

We note that it is necessary that subroutine BODY define the arrays XFOIL(I), FU(I), FL(I), FXU(IC), and FXL(I) which are, respectively, the x-ordinates of the input x-mesh XIN(I) which fall on the airfoil, the y-ordinates of the airfoil upper and lower surface divided by the thickness ratio, and the slopes of the airfoil upper and lower surface divided by the thickness ratio, all calculated at the XFOIL(I) points.

Flow environment option: To specify the type of flow environment; that is, free air or tunnel simulation conditions, the integer index BCTYPE must be set. Six different choices have been incorporated in the program according to the following schedule:

BCTYPE	Flow Condition
1	Free air
2	Solid wall tunnel
3	Free jet
4	Ideal slotted wall tunnel
5	Ideal porous/perforated wall tunnel
6	General tunnel wall

The program default value is BCTYPE = 1.

Conservative differencing option: An option has been provided for the user to specify whether conservative or nonconservative differencing be used at shock waves. This choice is made according to the logical index FCR:

FCR	Shock Difference Type
.T.	Fully conservative
.F.	Nonconservative

Program default value is FCR = .T.

Mesh refinement option: A convergence acceleration feature has been incorporated in the program which provides the user the option to initiate the iterative solution on a mesh that is coarser than the input mesh (fine mesh) on which the solution is ultimately desired. Provision has been made for two automatic grid reductions (medium and coarse) by the program. Since the grid change is accomplished by the deletion of every other mesh point, the following simple rules should be followed in specifying the input (fine) grid to take full advantage (i.e., two cuts) of this feature.

These rules consist of making the following grid point differences a multiple of four,

IMAXI - ITE
ITE - IMIN
JMAXI - JUP + 1
JLOW - JMIN + 1

which insure that each of the above four grid intervals can be halved twice. See Section 5.1 under COM 1 for a description of these indices.

The control for mesh refinement is made by specification of the integer index ICUT according to:

ICUT	Comment
0	Input mesh used to convergence
1	Input mesh may be cut once
2	Input mesh may be cut twice

We note that for the cases ICUT = 1 or 2, if the grid criteria given above are not met, the program will try to adjust if possible the grid by adding (never deleting) points. However, since the stated conditions are so mild, it is recommended that they be followed directly if the grid refinement option is desired. Program default value is ICUT = 2.

Intermediate printout option: If the user wished to obtain intermediate information concerning the convergence of the solution, an option is provided which will cause a limited printout to occur each time a specified number of iteration sweeps through the flow field are completed. Output includes iteration count, lift and moment coefficients, iteration error in circulation, and absolute values and locations of the maximum iteration error in ϕ, the residual $R_{j,i}$, and the upper and lower surface pressure coefficients. Control is through specification of the integer index IPRTER which causes output print every IPRTERth iteration sweep through the field. Program default value is IPRTER = 0.

Kutta condition enforcement option: If, in lieu of enforcing the Kutta condition, the user wishes to specify instead the lift coefficient, an option has been provided in the program to accommodate this. Control for this option is through the logical index KUTTA where

KUTTA	Comment
.T.	Kutta condition enforced
.F.	Lift coefficient specified; trailing-edge pressure discontinuous

Program default value is KUTTA = .T.

Physical or transonic I/O scaling option: A user choice for the type of scaling used in the input/output is available through the logical index PHYS. The I/O can be in either the usual physical scaled form (PHYS = .T.); that is, lengths normalized by airfoil chord, c, velocities by free-stream velocity, U_∞, pressure by free-stream dynamic pressure, q_∞, etc., or in the appropriate transonically scaled form (PHYS = .F.). If transonic scaling is used, we note the following requirements: the lateral coordinate of the input grid must be scaled according to

$$\tilde{y} = \delta^{1/3} M_\infty^m y$$

the angle of attack by

$$\tilde{\alpha} = \alpha/(\delta \cdot 180^\circ/\pi)$$

the half-tunnel height by

$$\tilde{H} = \delta^{1/3} M_\infty^m H$$

the lift coefficient CLSET by

$$(\mathrm{CL\tilde{S}ET}) = \mathrm{CLSET}/\delta^{2/3} M_\infty^{-n}$$

the porosity by

$$\tilde{P} = P/\delta^{1/3} M_\infty^m$$

the velocity potential by

$$\tilde{\phi} = \delta^{2/3} M_\infty^{-n} \phi$$

and instead of specifying the Mach number, M_∞, the similarity parameter K must be given, where

$$K = \frac{1 - M_\infty^2}{M_\infty^{2m}\, \delta^{2/3}}$$

and where the exponents (m,n) depend upon the choice of scaling and are defined in Table 4-I. Program default value is PHYS = .T.

Flow field printout option: In the final printout of a completed calculation, the user has the choice of obtaining flow field information [i.e., pressure coefficient, flow angle, and either local Mach number (PHYS = .T.) or local similarity parameter (PHYS = .F.)], either throughout or in a limited region of the flow field. This is controlled through the integer index PRTFLO according to

PRTFLO	Flow Field Printout
1	None
2	All
3	Three J lines about $\lvert\phi^{n+1}_{j,i} - \phi^{n}_{j,i}\rvert_{max}$ line

Program default value is PRTFLO = 1.

Result storage option: If at the completion of a calculation the user wishes, in addition to obtaining the printed output, to store the results on tape for future use, an option has been provided whereby the program will write the following information on tape 3:

(1) run title

(2) grid dimensions; IMAXI, JMAXI, IMIN, JMIN

(3) C_L, M_∞, α, δ, airfoil volume, far field doublet strength

(4) X and Y grid arrays

(5) $\phi_{j,i}$

Control is through the logical index PSAVE according to

PSAVE	Storage
.T.	Results stored on tape 3
.F.	No storage

Program default value is PSAVE = .F.

Initialization of ϕ option: Three choices are available for initializing the ϕ array to start the calculation. Selection is made according to the integer index PSTART where

PSTART	Initialization of ϕ
1	Set to zero
2	Read from tape 7
3	Used from previous case

Program default value is PSTART = 1.

Transonic similarity definition option: Four choices are available to the user for the specification of the particular transonic similarity form of the small-disturbance equation to be solved. The first three of these choices (SIMDEF = 1, 2, or 3) have been incorporated into the program and are detailed below. The fourth (SIMDEF = 4) is an open (inactive) option, requiring user-supplied definitions of the various scaling parameters calculated in subroutine SCALE.

The three active options reflect the different scaling rules of Cole, Spreiter, or Krupp and depend essentially upon the choice of values for the exponents (k,m,n) in the following representations of the small-disturbance perturbation velocity potential equation, stretched-lateral coordinate, and transonic similarity parameter; that is,

$$\left[(1 - M_\infty^2)\Phi_x - \frac{\gamma + 1}{2} M_\infty^k \Phi_x^2\right]_x + \Phi_{yy} = 0$$

or

$$\left[K\phi_x - \frac{\gamma + 1}{2} \phi_x^2\right]_x + \phi_{\tilde{y}\tilde{y}} = 0$$

where

$$\Phi(x,y) = \delta^{2/3} M_\infty^{-n} \phi(x,\tilde{y})$$

$$y = \delta^{-1/3} M_\infty^{-m} \tilde{y}$$

$$K = \frac{1 - M_\infty^2}{M_\infty^{2m} \delta^{2/3}}$$

The values of the exponents (k,m,n) for each of the three cases are given in Table 4-I. Selection of the particular scaling rule according to the value of SIMDEF is

SIMDEF	Scaling
1	Cole
2	Spreiter
3	Krupp
4	User defined

Program default value is SIMDEF = 3.

Default grid option: Two different sets of default grids have been incorporated in the program. These are the Krupp mesh defined in reference 2 and an analytical mesh based on a modified version of the grid discussed in reference 11. The program default x-mesh is the Krupp x-mesh XKRUPP(I), given in reference 2, while two default y-meshes are incorporated; that is, the Krupp y-mesh for free air calculations YFREE(I) and a modified y-mesh YTUN(I) for tunnel simulations. To use the analytical mesh, control is through the logical variable AMESH according to

AMESH	Grid Specification
.T.	Analytical mesh used
.F.	Either Krupp or user-specified mesh used

Program default value is AMESH = .F.

Input grid option: If the user wishes not to employ the default grids incorporated in the program, an option has been provided whereby an arbitrary mesh can be conveniently supplied to the program. Control is through the logical variables XGRDIN and YGRDIN according to

XGRDIN	Grid Specification
.T.	x-mesh provided by user in input data
.F.	Default x-mesh used

YGRDIN	Grid Specification
.T.	y-mesh provided by user in input data
.F.	Default y-mesh used

When the grids are user supplied, each element of the grid is defined in the input data, as illustrated in section 4.2.4. We note that the program default x-mesh is the Krupp x-mesh XKRUPP(I) defined in reference 2, while two default y-meshes are incorporated; that is, the Krupp y-mesh for free air calculations YFREE(I) and a modified y-mesh YTUN(I) for tunnel simulations. Program default values are XGRDIN = .F. and YGRDIN = .F.

4.2.3 Default values.- Each input variable has been assigned a default value that is stored in the BLOCK DATA subroutine. Consequently, unless the user wishes to change a variable from its default value, it is not necessary to enter it in the input data block. For convenience, the default values of all the input variables are tabulated below:

Variable	Default Value
AK	0.
ALPHA	0.12
AMESH	.F.
BCFOIL	3
BCTYPE	1
CLSET	0.
CVERGE	0.00001
DELTA	0.115
DVERGE	10.
EMACH	0.75
EPS	0.2
F	0.
FCR	.T.
GAM	1.4
H	0.
ICUT	2
IMAXI	77 (Krupp grid)
IMIN	1
IPRTER	10
JMAXI	56 (Krupp grid)
JMIN	1
KUTTA	.T.
MAXIT	500
NL	75 (Korn airfoil)
NU	100 (Korn airfoil)
PHYS	.T.
POR	0.
PRTFLO	1
PSAVE	.F.
PSTART	1
RIGF	0.
SIMDEF	3

Variable	Default Value
WCIRC	1.
WE(I), (I = 1,2,3)	1.8,1.9,1.95
XGRDIN	.F.
XIN(I)	XKRUPP(I)
XL(I)	(Korn airfoil)
XU(I)	(Korn airfoil)
YGRDIN	.F.
YIN(I)	YFREE(I) or YTUN(I)
YL(I)	(Korn airfoil)
YU(I)	(Korn airfoil)

It is important to realize that the above variables assume their default values only before the first case is run. Input variables, once user specified, are not changed internally by the program; thus, it is unnecessary to respecify them in subsequent cases if their values are to remain constant.

4.2.4 Input data format.- With the exception of a run title card, which is the first card of each case and on which the user can provide any alphanumeric information desired in columns 1 to 80, all of the input data for the program are read in under a NAMELIST format called INP. Consequently, the input format is most easily demonstrated by an example. Consider two cases of simulating the flow of air ($\gamma = 1.4$, default) past an NACA 0010 airfoil in an ideal slotted wall wind tunnel where the slot parameter is 0.07, and the tunnel half-height-to-chord ratio is 6.5. For the first case, the free-stream Mach number is 0.70, the angle of attack is 1.5°, and it is decided to match the measured tunnel lift coefficient of 0.187. The fully conservative equations (default) will be solved on the YTUN (tunnel default) y-grid and the basic Krupp XKRUPP (default) x-grid, employing the mesh refinement option with two grid halvings (default). The maximum number of iteration sweeps allowed will be 250, with intermediate printout desired after every fifth sweep, a convergence criterion on ϕ of 1×10^{-4}, a divergence criterion of 10. (default), with final flow field printout confined to three rows centered about the maximum iteration error in ϕ, and with the final results stored on tape 3. The transonic equation will be used in the similarity form as given by Spreiter, with the input/output in physical scaling terms (default), with the initial guess of ϕ equal to zero (default), and with the circulation relaxation factor equal to 0.9, the three elliptic relaxation factors equal to 1.3, 1.6, and 1.9 for the coarse, medium, and fine grids respectively. The pseudo-time term will be omitted from the differential equation (ϵ = 0.).

For the second case, all of the input parameters will be identical to the first except that the Mach number will be equal to 0.72 and the measured lift coefficient equal to 0.195.

Thus, the input data cards would read (note that with a NAMELIST format, input variable sequencing is arbitrary):

CARD NO. 1 - RUN TITLE CARD

COLUMN NO.

```
1                                                                              80
SAMPLE CALCULATION NO. 1
```

CARD NO. 2

COLUMN NO.

```
 2   6                                                                         80
 $INP EMACH=.7,F=.07,ALPHA=1.5,H=6.5,DELTA=.1,BCFOIL=1,
```

CARD NO. 3

COLUMN NO.

```
 2                                                                             80
 KUTTA=.F.,CLSET=.187,MAXIT=250,IPRTER=5,CVERGE=.0001,
```

CARD NO. 4

COLUMN NO.

```
 2                                                                             80
 PRTFLO=3,PSAVE=.T.,SIMDEF=2,WCIRC=.9,WE(1)=1.3,WE(2)=1.6,
```

CARD NO. 5

COLUMN NO.

```
 2                                                                             80
 WE(3)=1.9,EPS=0.,
```

CARD NO. 6

COLUMN NO.

```
 2                                                                             80
 $END
```

CARD NO. 7 - RUN TITLE CARD

COLUMN NO.

```
1                                                                              80
SAMPLE CALCULATION NO. 2
```

CARD NO. 8

COLUMN NO.

```
2   6                                                                     80
$INP EMACH=.72,CLSET=.195,
```

CARD NO. 9

COLUMN NO.

```
2                                                                         80
$END
```

CARD NO. 10

COLUMN NO.

```
1                                                                         80
FINISHED
```

We comment that the last card in the input data, following the final data card for the last case, must contain the word FINISHED in the first eight columns. This signals the program that there are no further cases to calculate and that the program is to stop.

If for the above cases the user wished to supply his own x- and y-grids, this would have been done by setting the grid input controls XGRDIN and YGRDIN equal to TRUE, specifying in the input the integers IMAXI and JMAXI, subject to the restriction IMAXI, JMAXI ≤ 100, and then identifying all of the IMAXI values of XIN(I) and JMAXI values of YIN(I) in the input data. For example, say that the user has an 81x60 (x,y) grid. Then, the card data to be included in the input for the first case would be (using some ficticious values for XIN, YIN):

COLUMN NO.

```
2                                                                         80
XGRDIN=.T.,IMAXI=81,XIN(1)=-3.0,XIN(2)=2.9,...
```

⋮ etc.

COLUMN NO.

```
2                                                                         80
XIN(80)=1.9,XIN(81)=2.0,YGRDIN=.T.,JMAXI=60,YIN(1)=-6.5,
```

⋮ etc.

COLUMN NO.

```
2                                                                         80
YIN(59)=6.4,YIN(60)=6.5,
```

A similar procedure would apply if the user wished to supply his own description of the airfoil surface; that is, arrays XL(I), YL(I), XU(I), and YU(I).

4.3 Error and Information Messages

The error and information messages printed by the program are listed in this section together with a brief explanation of why they occur and what to do when they are encountered. The first group of messages (1 to 12) is concerned with input quantities while the second group (13 to 21) occur during the execution of the program and involve certain limitations inherent in the methods used in this program.

(1) IMAX OR JMAX IS GREATER THAN 100, NOT ALLOWED

This message indicates that the input mesh size limitation IMAX, JMAX $\leq$ 100 has been violated.

(2) X MESH POINTS NOT MONOTONIC INCREASING

This message indicates an error in the input x-mesh that violates the condition $X_i > X_{i-1}$.

(3) Y MESH POINTS NOT MONOTONIC INCREASING

This message indicates an error in the input y-mesh that violates the condition $Y_j > Y_{j-1}$.

(4) MACH NUMBER NOT IN PERMITTED RANGE (0.5,2.0)

This message indicates that the input free-stream Mach number M_∞ is outside the nominally accepted transonic regime. A scheme solving the linear small-disturbance equation, rather than the nonlinear transonic equation solved in this program, should be employed.

(5) ALPHA NOT IN PERMITTED RANGE (-9.0,9.0)

This message indicates that the input angle of attack is beyond the range in which small-disturbance theory is considered valid; that is, $-9^\circ < \alpha < 9^\circ$.

(6) DELTA NOT IN PERMITTED RANGE (0.0,1.0)

This message indicates that the restriction on thickness ratio, $0. < \delta < 1.$, has been violated.

(7) AK = 0. VALUE OF AK MUST BE INPUT SINCE PHYS = .F.

This message indicates that the transonic similarity parameter $K = (1 - M_\infty^2)/M_\infty^{2m}\delta^{2/3}$ has not been input as is required when transonic rather than physical scaling; that is, PHYS = .F., is used in the input/output.

(8) MACH NUMBER TOO CLOSE TO 1., NOT ALLOWED

This message indicates that the input free-stream Mach number M_∞ is too close to one; that is, $|M_\infty - 1.| < 0.001$. This is unallowed since the program incorporates a representation of the far field solution containing the factor $1/\sqrt{K} \sim 1/|M_\infty^2 - 1|^{1/2}$. To correct this a special sonic free-stream far field solution would be required.

(9) ABNORMAL STOP IN SUBROUTINE BCEND
BCTYPE = 6 IS NOT USABLE

This message indicates to the user that the general tunnel wall boundary condition BCTYPE = 6 is an inactive option for which conditions have not yet been worked out.

(10) ABNORMAL STOP IN SUBROUTINE BODY
BCFOIL = 4 IS NOT USABLE

This message indicates to the user that the airfoil geometry specification BCFOIL = 4 is an inactive option requiring user definition. For details on necessary user input, see section 4.2.2.

(11) ABNORMAL STOP IN SUBROUTINE FARFLD
BCTYPE = 6 IS NOT USABLE

See message (9).

(12) ABNORMAL STOP IN SUBROUTINE SCALE
SIMDEF = 4 IS NOT USABLE

This message indicates to the user that the transonic similarity specification SIMDEF = 4 is an inactive option requiring user definition. For details on necessary user input, see section 4.2.2.

(13) NOTE - ONE OR MORE SHOCKS EXTEND OUTSIDE OF CONTOUR
CDWAVE DOES NOT EQUAL TOTAL WAVE DRAG

This message indicates that some of the shock waves present for the case at hand extend beyond the appropriate drag contour incorporated in the program, so that the wave drag calculated by the program does not equal the total wave drag. For details on the drag contours available, see the description of subroutine CDCOLE in section 5.2.

(14) SHOCK WAVE ATTACHED TO BODY. MOMENTUM INTEGRAL CANNOT
BE DONE. DRAG OBTAINED FROM SURFACE PRESSURE INTEGRAL.

This message indicates that, for reasons explained in section 5.2 under subroutine CDCOLE, the momentum integral method for calculating drag cannot be used. The drag is then calculated by integrating the surface pressures on the airfoil.

(15)

```
* * * * *  CAUTION  * * * * *
SOLUTION MAY BE INVALID
DETACHED SHOCK WAVE IS TOO CLOSE TO UPSTREAM BOUNDARY.
MOMENTUM INTEGRAL CANNOT BE DONE.  DRAG OBTAINED FROM
SURFACE PRESSURE INTEGRAL.
```

This message indicates that the bow shock is too close (within three mesh points) of the upstream boundary for the momentum integral along the bow shock to be performed, and that the drag is obtained by integrating the surface pressures on the airfoil. Also, because the bow shock is so close to the upstream boundary, the solution may be inaccurate and it is recommended that the upstream boundary be displaced further from the airfoil and the calculation repeated.

(16)

```
THE INPUT MESH CANNOT BE ADJUSTED FOR CUTOUT, BECAUSE
 IMAX OR JMAX IS TOO CLOSE TO THE LIMIT OF 100.
```

This message is written by subroutine CKMESH and indicates that, because the input mesh requirements for grid cutting described in section 4.2.2 under Mesh Refinement Option were not met, subroutine CKMESH has attempted to correct this by adding points. However, either the input X or Y grid is too close to the limit of 100 to allow this, with the result that no points are added, the mesh refinement option is bypassed, and the input mesh is used to convergence.

(17)

```
ABNORMAL STOP IN SUBROUTINE DROOTS.
                                     (0)
NONCONVERGENCE OF ITERATION FOR ALPHA(1)
                                     (2)
```

This message identifies one of the constants ALPHA0, ALPHA1, or ALPHA2 required in the subsonic far-field doublet solution in an ideal slotted wind tunnel, and indicates that the iterative solution for that constant has failed to converge. This error message should not normally occur. If it does, the input values for the tunnel slot parameter F and wall porosity factor P should be checked. See subroutine DROOTS in section 5.2 for details.

```
(18)  * * * * *  CAUTION  * * * * *
      SONIC LINE HAS REACHED A BOUNDARY
      THIS VIOLATES ASSUMPTIONS USED TO DERIVE BOUNDARY CONDITIONS.
      SOLUTION IS PROBABLY INVALID
```

This message indicates that the sonic line has extended out to the top and/or bottom grid boundaries, and that the solution is probably invalid since the assumptions under which the boundary conditions were derived have been violated. For free-stream flows, the difficulty can be remedied by moving the lateral boundaries out further, while for tunnel simulation calculations, this condition indicates that the flow has become choked. For the latter case, the details have not yet been worked out.

```
(19)  * * * * *  CAUTION  * * * * *
      NUMBER OF SONIC POINTS EXCEEDED 200. ARRAY DIMENSION
      EXCEEDED.  EXECUTION OF SUBROUTINE MILINE TERMINATED
```

This message indicated that the number of sonic points has exceeded 200 and subroutine MILINE, which determines the locations of the sonic lines, has terminated calculation. For the maximum grid size (100x100) allowed in this program, this error message should not occur. If it does, the remainder of the output should be checked for unusual behavior.

```
(20)  * * * * *  CAUTION  * * * * *
      MAXIMUM MACH NUMBER EXCEEDS 1.3
      SHOCK JUMPS IN ERROR IF UPSTREAM NORMAL MACH NUMBER GREATER
      THAN 1.3
```

This message is written by subroutine PRINT1 and indicates that the maximum Mach number along the dividing streamline, which includes the airfoil upper and lower surface, has exceeded 1.3. Consequently, if the normal Mach number just upstream of the shock wave is greater than 1.3 (this can be checked from the output of PRINT1) the shock jumps, as calculated by the program using isentropic small-disturbance theory, will be in error with the true Rankine-Hugoniot value since the usually accepted upper bound for validity of the isentropic assumption has been exceeded.

```
(21)  ABNORMAL STOP IN SUBROUTINE VROOTS
                                        (0)
      NONCONVERGENCE OF ITERATION FOR BETA(1)
                                        (2)
```

This message identifies one of the constants BETA0, BETA1, or BETA2 required in the subsonic far-field vortex solution in an ideal slotted wind tunnel, and indicates that the iterative solution for that constant has failed to converge. This error message should not normally occur. If it does, the input values for the tunnel slot parameter F and wall porosity factor P should be checked. See subroutine VROOTS in section 5.2 for details.

4.4 Program Output

The output generated by the program is described in this section. A general description of the format is provided in the first subsection, followed by a subsection containing several sample cases for user reference.

4.4.1 General description of the output.- The output format of the program is as follows. On the first page a header is printed giving the program name, some general information concerning the problem solved and method used, and identification of the program's authors. On the next page, all of the input cards for the entire run sequence are printed. On the first line of the third page, the information input by the user on the first run title card is printed. Next, all of the input variables described in section 4.2.1 are printed. This includes, of course, variables not specified in the input data and which then assume their default values. On the fourth page the airfoil geometry information required by the program is printed. First, a header is printed to inform the user whether the printout is in physical variables (PHYS = .T.), in which case all of the airfoil geometric characteristics are normalized by the chord; or if the printout is in transonic similarity variables (PHYS = .F.), for which the geometric characteristics are normalized by chord length times thickness ratio. Next, the maximum thickness, airfoil volume (per unit span), and maximum camber are output. Finally, the ordinates and slopes of the upper and lower surface, together with the thickness and camber distributions are provided in tabular form at the input x-mesh points. On the fifth page, intermediate output information for the coarse mesh (if a coarse grid is used) is provided. This includes the elliptic relaxation factor, the value of ϵ (EPS) used, and the maximum number of iterations for this mesh. This is followed by a heading which contains, from left to right, the iteration count (ITERATION), lift coefficient (CL), moment coefficient (CM), I and J locations (IERR,JERR) where the maximum iteration error in ϕ occurs, the absolute magnitude of the maximum iteration error in ϕ (ERROR), the I and J locations (IRL,JRL) of the maximum residual, the absolute magnitude of the maximum residual (BIGRL), the absolute magnitude of the iteration error in circulation (ERCIRC), the I location (ICPU) of the point on the airfoil where the maximum iteration error in upper surface pressure coefficient occurs, the absolute magnitude of the maximum iteration error in upper surface pressure coefficient (CPERRU), and the corresponding results (ICPL,CPERRL) for the maximum iteration error in surface pressure coefficient. A printout of these quantities is provided at every IPRTERth iteration until the solution either converges or diverges, or the iteration limit is reached; at this point, a message is printed indicating which of these three alternatives occurred. On the sixth page, the final output for the coarse mesh is given. This consists of the final coarse-mesh values of the lift and moment coefficients and tabulated values of the pressure coefficient and local Mach number (or local similarity parameter if PHYS = .F.) along the entire length of the dividing (y = 0) streamline, including both the upper and lower surface of the airfoil. In addition, the x- and y-coarse-mesh ordinates are also provided. The format is as follows. First, the lift and moment coefficients are given, together with the value of the critical pressure coefficient. Next, a header is printed which contains, from left to right, the I index of the current x-mesh point (I), the x value of the mesh point (X), the pressure coefficient (CP), and local Mach number (M1) just below the dividing streamline with the additional title $\left(\begin{matrix}\text{LOWER}\\ y = 0-\end{matrix}\right)$, and then the pressure coefficient (CP) and local Mach number (M1) just above the dividing streamline with the additional title $\left(\begin{matrix}\text{UPPER}\\ y = 0+\end{matrix}\right)$. Tabulated values of these quantities follow and are provided for each x-mesh point of the coarse grid. Locations of the airfoil leading and trailing edges are indicated in the output. Alongside the table of values, a concomitant printer plot of the tabulated pressure coefficients is given. Finally, all the y-values of the coarse grid y mesh are printed. On the seventh and eighth pages, the intermediate and final output, respectively, for the medium grid (if a medium grid is used) are provided in the same format as described above for the coarse grid. On the eight page, the intermediate output for the fine mesh is given. On the following page, the final print output begins for the fine mesh. On this page, information is provided informing the user whether the printout is in physical or transonic variables, and if it is in physical variables which particular definition of similarity parameters was used. Also indicated is the type of boundary condition specified (free air or type of wind tunnel), whether the difference equations were in conservative form or not, and whether the Kutta condition was enforced or the lift coefficient specified. Next, some limited airfoil geometry and flow field information is given (i.e., free-stream Mach number and airfoil thickness ratio if physical scaling is used, (PHYS = .T.), plus angle of attack and transonic similarity parameter in all cases, and far-field doublet strength and airfoil volume per unit span when $M_\infty < 1$. Finally, if PHYS = .T., the transonic scaling parameters used by the program to scale the pressure, lift, drag, and moment coefficient, the y-ordinate, and the flow angle are printed. On the next page, the fine grid lift and moment coefficients, pressure coefficients and local Mach number on the dividing streamline together with the x- and y-fine mesh ordinates, and a printer plot of the pressure coefficients is given in the format previously described for the coarse grid. On the next page, a flow character map is provided. Each point in the mesh is characterized according to whether it is a parabolic point (P), hyperbolic point (H), shock point (S), or an elliptic point (—). The following page provides a Mach number map of the flow field. Each grid point receives either a number or letter letter designation according to the symbol table provided in the plot. On the next page, a laterally expanded printer plot of the pressure coefficients on the dividing streamline and airfoil surface is provided. For wind-tunnel wall simulation calculation (i.e., BCTYPE ≠ 1 or 3), tabulated values of the pressure coefficient and flow angle along the entire length of the wind-tunnel walls ($y = H\underline{+}$) are given next. The format is similar to that provided for output of the pressure distributing along the dividing streamline discussed previously. This consists of tabulated values of the pressure coefficient (CP) and the flow angle (THETA) along the bottom $\left(\begin{matrix}\text{LOWER}\\ y = -H\end{matrix}\right.$ and top $\left.\begin{matrix}\text{UPPER}\\ y = +H\end{matrix}\right)$ surfaces of the wind tunnel at each of the fine mesh x-grid (X) locations. A printer plot of the pressure coefficients is also provided alongside the tabulated values. On the next two pages, output describing the sonic lines is provided. The first page gives tabulated values of the sonic line coordinates, while the second page provides a printer plot of the sonic lines throughout the flow field. On the following page, flow field output, if requested by the user, is printed. This consists of tabulated values of the pressure coefficient, flow angle, and local Mach number (or local similarity parameter if PHYS = .F.) along the entire length of y = constant lines on the fine mesh. The format is as follows. First, a header is printed which contains, in the first two columns, the I index of the current x-mesh point (I) and the x value of the x-mesh point (X) The next nine columns contain three sets of pressure coefficients (CP), flow angles (THETA), and local Mach numbers (M1) along three different y = constant rows, with an additional title above each of the sets indicating both the

J index and associated y-value of the row. Tabulated values of these quantities follow for each x-mesh point of the fine grid. If the flow field print option PRTFLO = 3 was chosen, the flow field output is confined to three J lines about the location of maximum iteration error in ϕ, and all the flow field output is contained across one page. However, if the entire flow field is to be printed (PRTFLO = 2), then the above format is repeated for three successive rows at a time until the entire field is covered. Following this, the shock wave drag profiles and total pressure loss profiles are given. For each shock wave within the momentum contour, the total wave drag is given together with tabulated values of the shock wave drag (CD(y)) and the total pressure (PO/POINF) as functions of the lateral coordinate (y). On the next page, information regarding the drag coefficient is given. If the drag was calculated by the momentum integral method, the locations of the contour boundaries are given together with the individual contributions to the drag from the integrations of the drag integral along each of these boundaries. In addition, information regarding the number of shock waves inside the contour and the total wave drag are printed. Should any of the shocks extend beyond the momentum contour, a message is printed indicating this and the fact that the wave drag calculated does not equal the total wave drag. Finally, the total of all the separate drag contributions is given. If the momentum integral method could not be used to calculate the drag--either because the bow shock was attached or too close to the nose or because the bow shock was too close to the upstream boundary--a message to that effect is printed and the drag is then calculated by an integration of surface pressures. The time to run the case in seconds is provided on the next page. This would conclude the output for an individual case. The above format is repeated until all cases are run. Following this, a final page of output is printed which consists simply of the word FINISHED, indicating a normal exit from the program.

4.4.2 Sample cases.- In order to provide reference checks on the programs, six sample test cases have been run exercising a representative sample of the various options available. The results are provided in figures 4-4 through 4-9. In each case the input data are provided together with the corresponding output. Because of the large amount of output information, full program output is provided only for the first two cases (figs. 4-4,4-5); however, the abbreviated output given for the remaining cases is sufficient to verify the correctness of the program under those options selected.

4.5 Comparisons with Data and Other Methods

A variety of comparisons of the results predicted by the methods employed in this program have been presented in the literature (refs. 1, 2, 3, 6, 7, 8, 9, and 20). For convenience, several of those comparisons are briefly repeated here in order to provide the user both with a general assessment of the accuracy of the methods used and some idea of the typical solution behavior that can be expected. For further information, the user is referred to the excellent review article by Bailey (ref. 21).

Basically, the comparisons presented here fall into three categories; that is, (1) comparisons with exact (hodograph) inviscid solutions, (2) comparisons with more accurate (full inviscid, full potential) numerical solutions, and (3) comparisons with data. The first two categories serve to assess the accuracy of the numerical technique and the small-disturbance potential approximation, while the third serves to demonstrate the influence of tunnel interference and viscous effects.

In figure 4-10 a comparison is made of the small-disturbance relaxation solutions calculated by the Krupp-Murman techniques (ref. 3) employed in this program with the exact hodograph solutions (ref. 22) for a NLR quasi-elliptical airfoil at $\alpha = 1.32^\circ$, $M_\infty = 0.7557$. A similar result is shown in figure 4-11 for a Garabedian-Korn airfoil at $\alpha = 0.12^\circ$, $M_\infty = 0.76$. Also provided in those two figures are relaxation solutions (refs. 23 and 24) for the full potential equation. We note that in both of those cases, the small-disturbance relaxation solutions and the full potential relaxation solutions show a weak shock as compared to the exact hodograph solution, which is for the shock-free design condition. For the Garabedian-Korn airfoil, the small-disturbance solution also predicts a somewhat smaller expansion beyond the nose. The overall agreement, however, between the small-disturbance solutions and the exact and full potential solutions is quite good, particularly in light of the fact that flows past such shock-free airfoils are extremely sensitive to small changes in geometry and flow conditions, and therefore provide an exacting test of the accuracy of any transonic predictive method. Figure 4-12 shows a further comparison of the small-disturbance results predicted by the Krupp-Murman method and a relaxation solution of the full inviscid equations for a subcritical flow (ref. 25). Again, the agreement is very good, even at this low ($M_\infty = 0.63$) a Mach number.

The results presented in the preceding three figures are for shock free flows. For flows containing shock waves, the relaxation methods used above for the full equations do not employ conservative difference operators and consequently do not satisfy the proper shock jump conditions. Some idea of the error involved in solving the exact isentropic equation in nonconservative form (ref. 26) and the calculation of the isentropic solution using a conservative time-dependent procedure (ref. 27) is given in figure 4-13. Also provided is the small-disturbance result predicted by the present program when using the fully conservative (FCR) option. We note that the shock location predicted by the two conservative calculations is essentially in agreement, while the nonconservative result predicts a weaker shock displaced forward of the conservative location. We note that the differences in shock jump values observed above between the two conservative calculations is not due to an accuracy problem in the above computations but is attributable to the difference in pressure jump between the isentropic approximation to the Rankine-Hugoniot relations, which is the jump condition satisfied in the fully conservative time-dependent calculation, and the small-disturbance approximation to the R-H conditions, which is the corresponding shock jump condition satisfied when using the conservative (FCR) option in this program.

Comparisons between the fully conservative (FCR) and not fully conservative (NCR) small-disturbance solutions predicted by the present finite-difference scheme are of interest in order to assess the differences between such calculations. Such results have been discussed previously in reference 19, and figures 4-14 and 4-15 summarize the essential conclusions of that study and provide a further comparison with a more exact result. Figure 4-14 compares the FCR and NCR solutions for the similarity surface pressure distributions for a subsonic free-stream supercritical flow as the finite-difference grid is refined. We note that while the FCR solution approaches the theoretical, small-disturbance shock jump, the NCR solution does not. Figure 4-15 shows the corresponding result for a supersonic free stream and a detached bow wave, and also includes the result of a conservative time-dependent calculation for the exact isentropic equations (ref. 28). While the FCR and the time-dependent method produce essentially the same result, the NCR method yields too great a detachment distance. Furthermore, it was found (ref. 19) that the NCR method predicts shock jumps which are considerably in error for strong oblique shocks, although the results for normal and weak oblique shocks remain reasonable. We note, as pointed out previously (refs. 1 and 21), in many cases of $M_\infty < 1$ transonic flows, the NCR predicted surface pressure distributions are usually in better agreement with experimental data than the FCR results, due to boundary-layer weakening of the shock wave. Unfortunately, this agreement is clearly fortuitous, and because the NCR method does not yield a unique solution nor account for viscous effects in any rational manner, it is not recommended for general use.

In the absence of shock wave-boundary layer separation, it has been suggested (refs. 19 and 29) that the principal viscous effect to account for is the reduction in circulation from the inviscid value determined by the Kutta condition. This effect is illustrated in figure 4-16 where the data from reference 15 is compared with the two FCR small-disturbance solutions found by: (a) specifying the Kutta condition and (b) fixing the circulation at the value determined by the experimental lift. Considerably better agreement with data results in the latter case except, of course, for points near the trailing edge.

Finally, some indication of the effects of wind-tunnel interference at transonic speeds is given in figures 4-17 and 4-18. Figure 4-17 shows the FCR solutions for an NACA 0012 airfoil at $\alpha = 0^\circ$ in free air and for an ideal slotted wall compared to the data taken in the RAE slotted wind tunnel (ref. 30). A forward movement of the shock due to tunnel interference is indicated. This effect is again shown by the results presented in figure 4-18, where data taken from the NASA/Ames Research Center 2- by 2-Foot Wind Tunnel (ref. 1) is compared with the FCR solutions for free air, an ideal slotted tunnel, and an ideal slotted tunnel with the lift fixed at the experimental value. In this example, the effects of wind-tunnel interference on lift and shock position are of the same magnitude as the viscous effects as evident by their respective changes from the free air to the slotted tunnel Kutta condition result (interference effect), and from the slotted tunnel Kutta condition result to the slotted tunnel fixed C_L result (viscous effect).

4.6 Program Run Times

The development of the structure of the present code as well as the actual debugging process have been carried out on a Control Data Corporation 7600 computer. Therefore, the majority of run time information and experience is for that machine. However, the most crucial time information required by the general user is simply the CPU time required for one iteration sweep of the flow field on a typical fine grid, since this will provide directly an estimate of the maximum run time required for a particular case. The resolution of the default free air grid (77×56) and default tunnel grid (77×48) used in the program is generally sufficient for most engineering applications, so that the time per iteration sweep on these grids should provide a fairly general and useful result. We have determined that the sweep time per iteration on these grids using the FTN, OPT = 2 compiler on the CDC 7600 is approximately 0.04 sec/sweep. Since the medium and coarse meshes contain approximately one-fourth and one-sixteenth as many points, the approximate maximum CPU execution time per case when employing the grid refinement option is

$$\text{Maximum CPU Execution Time} \doteq \text{MAXIT}\cdot(1 + 1/4 + 1/16)\cdot(0.04)\,\text{sec}$$

(CDC 7600; FTN, OPT = 2)

$$\doteq 0.0525\cdot\text{MAXIT sec}$$

The analogous result for the CDC 6600 is

$$\text{Maximum CPU Execution Time} \doteq 0.310\ \text{MAXIT sec}$$

(CDC 6600; FTN, OPT = 2)

We note that on the analytical mesh (81×64), the corresponding CDC 7600 time is approximately 0.055 sec/sweep.

With regard to improving run times, numerical experimentation has shown that by using the mesh refinement option execution times can be reduced by 30 percent to 60 percent in most cases, in comparison to the same calculation started and carried to completion on the fine grid alone. In addition, it was found that a number of cases which diverged when started on the fine mesh, converged smoothly on the final mesh when the mesh refinement option was used. Consequently, the additional programming required for this option appears well worthwhile in terms of enhanced stability and computational time. Finally, we note that a number of proven convergence acceleration techniques, such as the extrapolation methods used by Hafez and Cheng (ref. 31) have not yet been incorporated into the present version of the code. Future use of these improvements, as well as others currently under development--such as direct or semidirect solvers (refs. 32 and 33) and shock fitting (ref. 34)--can undoubtedly augment the methods used in the current program.

4.7 Machine Compatability

The present code has been written in FORTRAN and primarily tested out on a Control Data Corporation 7600 using the FTN, OPT = 2 compiler. Care has been taken in the programming to insure that the code can be easily adapted for use on other machines. All header output statements are written in the universal Hollerith format. Double precision arithmetic is not used. Storage requirements have been reduced as much as possible by the liberal use of common blocks to pass information internally. The code has been successfully check tested on a CDC 6600 and required no changes.

	k	n	m
Cole	0	0	0
Spreiter	2	2/3	2/3
Krupp	7/4	3/4	1/2

Table 4-I.- Exponents for transonic similarity parameters.

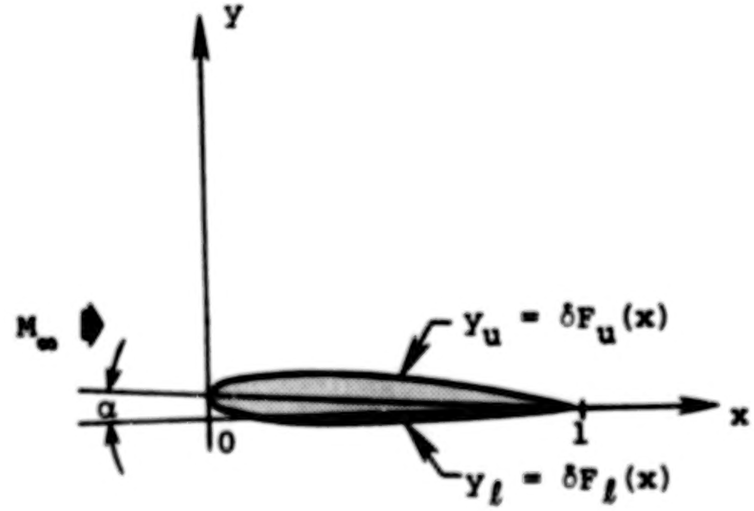
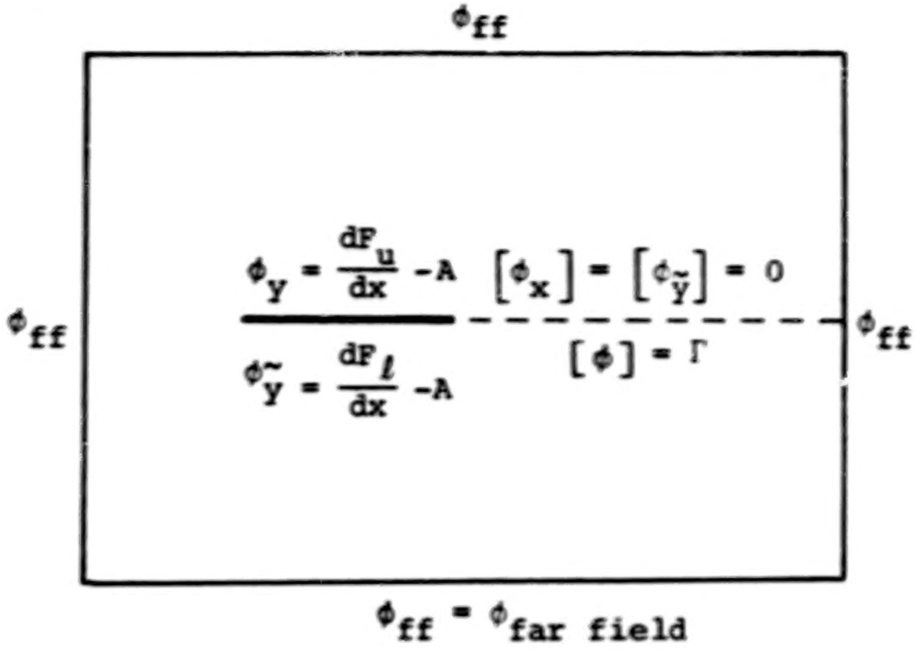
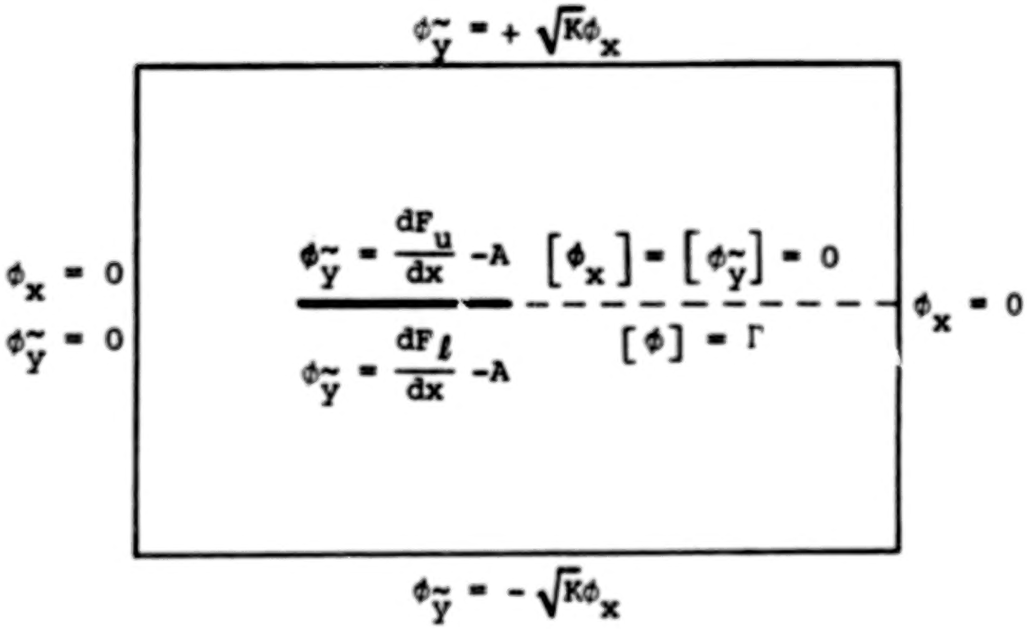

y
M_∞
$y_u = \delta F_u(x)$
α
0
1
x
$y_\ell = \delta F_\ell(x)$

Figure 4-1.- Coordinate system and airfoil geometry.

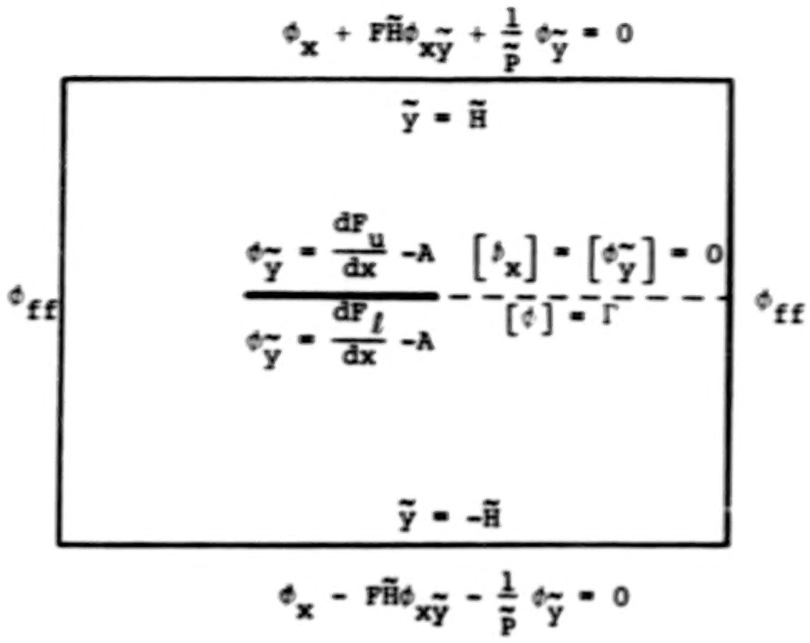

ϕ_{ff}
$\phi_y = \frac{dF_u}{dx} - A$
$[\phi_x] = [\phi_{\tilde{y}}] = 0$
ϕ_{ff}
ϕ_{ff}
$[\phi] = \Gamma$
$\phi_{\tilde{y}} = \frac{dF_l}{dx} - A$
$\phi_{ff} = \phi_{far\ field}$

(a) Subsonic free stream ($M_\infty < 1$).

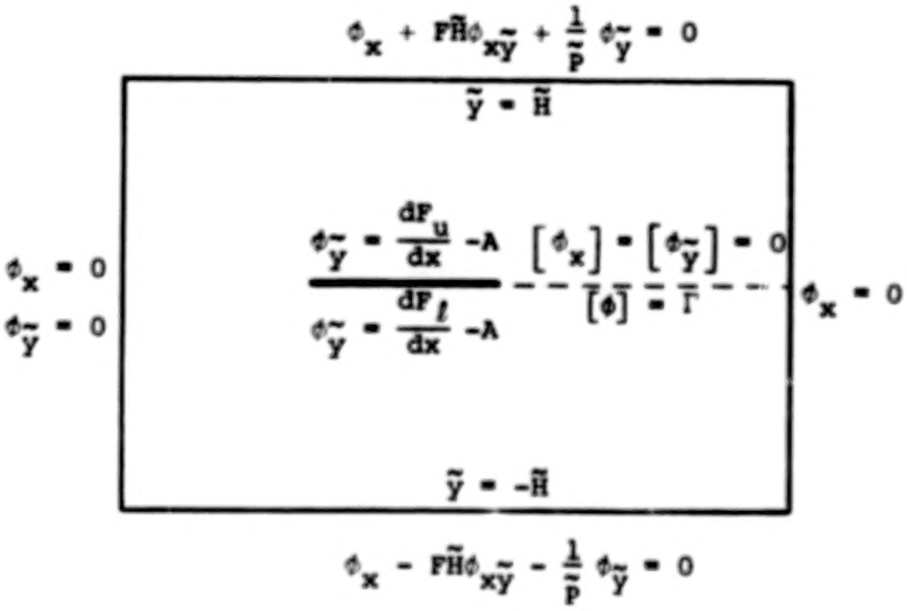

$\phi_{\tilde{y}} = +\sqrt{K}\phi_x$
$\phi_{\tilde{y}} = \frac{dF_u}{dx} - A$
$[\phi_x] = [\phi_{\tilde{y}}] = 0$
$\phi_x = 0$
$\phi_{\tilde{y}} = 0$
$\phi_x = 0$
$[\phi] = \Gamma$
$\phi_{\tilde{y}} = \frac{dF_l}{dx} - A$
$\phi_{\tilde{y}} = -\sqrt{K}\phi_x$

(b) Supersonic free stream ($M_\infty > 1$).

Figure 4-2.- Summary of boundary conditions for free air flows.


$\phi_x + F\tilde{H}\phi_{x\tilde{y}} + \frac{1}{\tilde{P}}\phi_{\tilde{y}} = 0$
$\tilde{y} = \tilde{H}$
$\phi_{\tilde{y}} = \frac{dF_u}{dx} - A$
$[\phi_x] = [\phi_{\tilde{y}}] = 0$
ϕ_{ff}
$[\phi] = \Gamma$
ϕ_{ff}
$\phi_{\tilde{y}} = \frac{dF_\ell}{dx} - A$
$\tilde{y} = -\tilde{H}$
$\phi_x - F\tilde{H}\phi_{x\tilde{y}} - \frac{1}{\tilde{P}}\phi_{\tilde{y}} = 0$

(a) Subsonic free stream ($M_\infty < 1$).


$\phi_x + F\tilde{H}\phi_{x\tilde{y}} + \frac{1}{\tilde{P}}\phi_{\tilde{y}} = 0$
$\tilde{y} = \tilde{H}$
$\phi_{\tilde{y}} = \frac{dF_u}{dx} - A$
$[\phi_x] = [\phi_{\tilde{y}}] = 0$
$\phi_x = 0$
$\phi_{\tilde{y}} = 0$
$[\phi] = \Gamma$
$\phi_x = 0$
$\phi_{\tilde{y}} = \frac{dF_\ell}{dx} - A$
$\tilde{y} = -\tilde{H}$
$\phi_x - F\tilde{H}\phi_{x\tilde{y}} - \frac{1}{\tilde{P}}\phi_{\tilde{y}} = 0$

(b) Supersonic free stream ($M_\infty > 1$).

Figure 4-3.- Summary of boundary conditions for wind-tunnel simulations.

```
SAMPLE CASE 1 - PROGRAM DEFAULT OPTION - KORN AIRFOIL
$INP
$END
```

(a) Input.

```
SAMPLE CASE 1 - PROGRAM DEFAULT OPTION - KORN AIRFOIL


                                       INPUT PARAMETERS
                                       ****************

  EMACH =   .75000     POR =  0.00000     IMIN =   1     BCTYPE = 1        AMESH = F
  DELTA =   .11500   CLSET =  0.00000    IMAXI =  77     BCFOIL = 3         PHYS = T
  ALPHA =   .12000     EPS =   .20000     JMIN =   1     PSTART = 1        PSAVE = F
     AK =  0.00000    RIGF =  0.00000    JMAXI =  56     PRTFLO = 1        KUTTA = T
    GAM =  1.40000   WCIRC =  1.00000    MAXIT = 500     IPRTER = 10         FCR = T
      F =  0.00000  CVERGE =   .00001       NU = 100     SIMDEF = 3
      H =  0.00000  DVERGE =     10.0       NL =  75       ICUT = 2
     WE = 1.80,1.90,1.95

 XIN
-1.075000     -.950000     -.825000     -.700000     -.575000     -.450000     [illegible]  [illegible]
 -.175000     -.125000     -.075000     -.052500     -.035000     -.022500     [illegible]  [illegible]
 -.002500      .002500      .007500      .012500      .017500      .022500      .027500      .032500
  .037500      .045000      .055000      .065000      .075000      .085000     [illegible]  [illegible]
  .140625      .171875      .203125      .234375      .265625      .296875      .328125      .359375
  .390625      .421875      .453125      .484375      .515625      .546875      .578125      .609375
  .640625      .671875      .703125      .734375      .765625      .796875      .828125      .859375
  .885000      .902500      .915000      .932500      .945000      .960000      .975000      .990000
 1.000000     1.010000     1.020000     1.035000     1.050000     1.150000     [illegible]  [illegible]
 1.400000     1.500000     1.625000     1.750000     1.875000

 YIN
-5.200000    -4.400000    [illegible]  -3.200000    -2.400000    -1.950000    -1.600000    -1.350000
-1.150000     -.950000     -.800000     -.650000     -.550000     -.450000     -.350000     -.300000
 [illegible]   -.270000     -.245000     -.219000     -.180000     -.150000     -.125000     -.100000
 -.075000     -.050000     -.030000     -.010000     [illegible]  [illegible]  [illegible]  [illegible]
  .100000      .125000      .150000      .180000      .210000      .240000      .270000      .300000
  .340000      .390000      .453333      .550000      .650000      .800000      .950000     1.150000
 1.350000     1.600000     1.950000     2.400000     3.200000     3.600000     [illegible]  [illegible]

 XU
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]                                         1.000000

 YU
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]                                         0.000000
```

Figure 4-4.- Sample test case 1 - program default option - subsonic, free air simulation (full output): Korn airfoil, δ= 0.115, α = 0.12°, M_∞ = 0.75.

```
 XL
0.000000    .000012    .000043    .000103    .000249    .000348    [illegible] [illegible]
 .001011    .001481    .001875    .002316    .003355    .004201    .004747    .005779
 .007035    .008265    .009909    .012286    .015346    .019276    .025335    .029374
 .039095    .052516    .062469    .073324    .085290    .099822    [illegible] .140967
 .167164    .202933    .228511    .247595    .263995    .282567    [illegible] [illegible]
 .324075    .344872    [illegible] .387544   .404492    .426308    [illegible] [illegible]
 .521837    [illegible] .578512   [illegible] .623479   .642152    [illegible] .674212
 .690340    [illegible] .726684   .746683    .768502    .784892    [illegible] [illegible]
 .838549    .956817    .879431    .903723    .926504    .943652    [illegible] .973523
 .988187    .996582   1.000000

 YL
0.000000   -.003760   [illegible] [illegible] [illegible] -.003806 [illegible] [illegible]
-.006133   -.007183   -.007933   -.008676   -.009776   -.011244   [illegible] [illegible]
-.013983   -.014962   -.016175   -.017636   -.019336   -.021258   [illegible] [illegible]
-.028634   -.032423   -.034840   -.037482   -.039456   -.041862   -.044483   [illegible]
-.049298   -.051443   -.052406   -.052459   -.053362   -.052117   [illegible] [illegible]
-.052562   -.051951   -.051218   -.050313   -.049004   -.047495   [illegible] [illegible]
-.038336   -.034916   -.031104   -.027333   -.024661   -.021854   [illegible] [illegible]
-.014527   -.011771   -.008228   -.006537   -.003868   -.002086   [illegible] [illegible]
 .002227    .003224    .003885    .004212    .004067    .003657   [illegible] [illegible]
 .001329    .000376   0.000000
```

```
                                        AIRFOIL GEOMETRY OUTPUT
                                        ***********************

PRINTOUT IN PHYSICAL VARIABLES NORMALIZED BY CHORD LENGTH                       MAX THICKNESS =    .11969193
AIRFOIL VOLUME=   .07956017                                                     MAX CAMBER    =    .01954304

 INPUT  GRID          UPPER  SURFACE                   LOWER  SURFACE
      X               Y            DY/DX              Y            DY/DX          THICKNESS     CAMBER

   .00250000       .00959171   1.45573113       -.00895547  -1.54309747        .00927044    [illegible]
   .00750000       .01470383    .61207528       -.01486528   -.66527524        .01478455    [illegible]
   .01250000       .01831665    .56637444       -.01776239   -.50753222        .01803942    [illegible]
   .01750000       .02132582    .55642563       -.02041959   -.46511405        .02087270    [illegible]
   .02250000       .02399963    .47646664       -.02267782   -.42225950        .02331842    [illegible]
   .02750000       .02631987    .44558493       -.02467419   -.37876104        .02549703    [illegible]
   .03250000       .02846698    .41156450       -.02648365   -.34495812        .02747462    [illegible]
   .03750000       .03044665    .37499980       -.02813392   -.31735308        .02929028    [illegible]
   .04500000       .03312592    .33599923       -.03030772   -.28473871        .03171682    [illegible]
   .05500000       .03627290    .29719151       -.03325353   -.24997657        .03476322    [illegible]
   .06500000       .03904111    .25543566       -.03541151   -.22262497        .03722631     .00181480
   .07500000       .04141318    .21765425       -.03751777   -.19949357        .03946548    [illegible]
   .08500000       .04339674    .18654845       -.03963423   -.17676220        .04151549    [illegible]
   .09750000       .04547125    .15571664       -.04155113   -.15725605        .04348617    [illegible]
   .11500000       .04766777    .13267017       -.04402275   -.13157246        .04584526    [illegible]
   .14062500       .05107517    .11107138       -.04698667   -.11053417        .04902792    [illegible]
   .17187500       .05422338    .09138735       -.04963096   -.07664961        .05192717    [illegible]
   .20312500       .05682572    .07566296       -.05145108   -.04617982        .05413840    [illegible]
   .23437500       .05897522    .06223396       -.05256752   -.02578214        .05577137    [illegible]
   .26562500       .06073949    .05053493       -.05307482   -.00710822        .05690716    [illegible]
   .29687500       .06214326    .03999612       -.05302871    .00999124        .05758599    [illegible]
   .32812500       .06323120    .02494726       -.05246513    .02617139        .05784800     .00538393
   .35937500       .06401751    .02066257       -.05134534    .04194736        .05770642    [illegible]
   .39062500       .06451653    .01156033       -.04984448    .05725378        .05718051     .00733602
   .42187500       .06473420    .00241416       -.04781989    .07224666        .05627705    [illegible]
   .45312500       .06467133   -.00656035       -.04533339    .08689782        .05500236    [illegible]
   .48437500       .06421053   -.01594653       -.04243644    .10059528        .05335847    [illegible]
   .51562500       .06366431   -.02603701       -.03905118    .11380738        .05135775    [illegible]
   .54687500       .06268375   -.03696603       -.03524270    .12645358        .04896327    [illegible]
   .57812500       .06133234   -.04979547       -.03117003    .13715576        .04625119    [illegible]
   .60937500       .05955242   -.06436061       -.02674112    .14591750        .04314677    [illegible]
   .64062500       .05724524   -.08046577       -.02205493    .15119819        .03965010    [illegible]
   .67187500       .05451646   -.09802562       -.01732781    .15267515        .03591913    [illegible]
   .70312500       .05117616   -.11046994       -.01261049    .14780410        .03189432    [illegible]
   .73437500       .04725652   -.13370925       -.00816933    .13993154        .02771293     .01954359
   .76562500       .04280623   -.15103101       -.00420143    .11693724        .02350383    [illegible]
   .79687500       .03783063   -.16679849       -.00091374    .09290191        .01937609    [illegible]
   .82812500       .03239200   -.18132322        .00157084    .06659416        .01541058    [illegible]
   .85937500       .02693266   -.19290638        .00324653    .04010531        .01184305    [illegible]
   .88500000       .02156184   -.19969862        .00406325    .01893465        .00874929    [illegible]
   .90000000       .01856618   -.20087864        .00419381    .00644729        .00718619    [illegible]
   .91500000       .01546642   -.20052560        .00419572   -.00623884        .00563535    [illegible]
   .93000000       .01249528   -.19797346        .00403511   -.01924826        .00423009    [illegible]
   .94500000       .00955711   -.19341345        .00361309   -.03322763        .00297201    [illegible]
   .96000000       .00676666   -.18613192        .00300379   -.04814109        .00188144    [illegible]
   .97500000       .00399207   -.17519895        .00215205   -.06622514        .00092001     .00307208
   .99000000       .00147612   -.15923679        .00102946   -.08737525        .00022333    [illegible]
  1.00000000      0.00000000   -.13691599       0.00000000   -.11008565       0.00000000   0.00000000
```

Figure 4-4.- Continued.

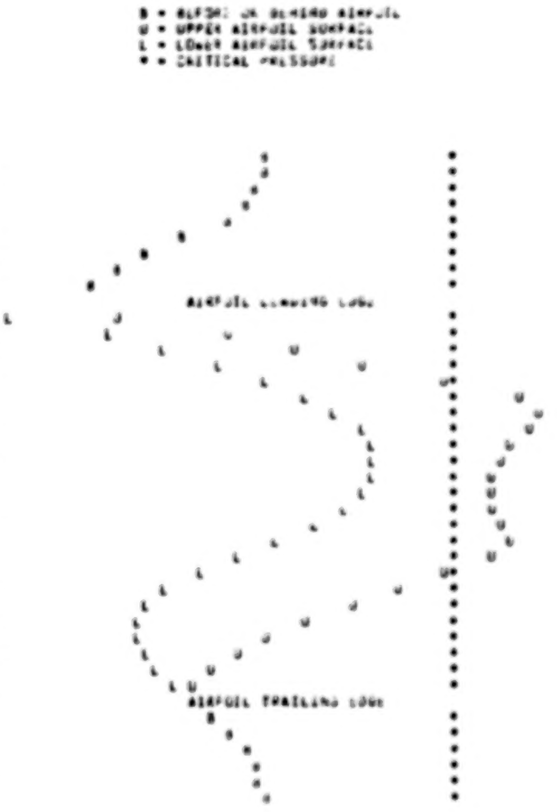
```
                                                INTERMEDIATE OUTPUT FOR COARSE MESH

       WE =   1.0000      EPS =     .2000      MAXIT FOR THIS MESH =  125

ITERATION     CL          CM      IERR  JERR     ERROR      IRL   JRL       BIGRL         ERCIRC    ICPU     CPERRU     ICPL     CPERRL

    10      .45126     -.16416     12     7    .4604E-01     6     7     .9957E+02     .4366E-01    12    .5703E+00    16    .3936E+00
    20      .52032     -.19065      5     4    .1936E-01     6     7     .1756E+03     .7888E-02    16    .9519E-01     6    .2603E+00
    30      .56601     -.19205      5     7    .7516E-02     6     7     .9360E+02     .5858E-02     8    .1324E+00    10    .3931E-01
    40      .59563     -.19427     10     7    .4266E-02     6     7     .4973E+02     .2151E-02     9    .7349E-01     7    .2037E-01
    50      .60322     -.19424      5     7    .2120E-02     6     7     .2546E+02     .1601E-02     9    .3169E-01     7    .1275E-01
    60      .60992     -.19472     11     7    .1166E-02     6     7     .1390E+02     .7924E-03     9    .1884E-01     7    .7320E-02
    70      .61343     -.19490      5     7    .6631E-03     6     7     .7928E+01     .5202E-03     9    .9779E-02     7    .2019E-02
    80      .61554     -.19512      6     7    .3044E-03     6     7     .4595E+01     .2905E-03     9    .5994E-02     7    .1701E-02
    90      .61670     -.19520      5     7    .2290E-03     6     7     .2606E+01     .1740E-03     9    .3547E-02     7    .1010E-02
   100      .61751     -.19525      5     7    .1310E-03     6     7     .1573E+01     .1014E-03     9    .2103E-02     7    .5931E-03
   110      .61793     -.19528      5     7    .7715E-04     4     7     .9203E+00     .5952E-04     9    .1241E-02     7    .3496E-03
   120      .61818     -.19530      5     7    .4921E-04     6     7     .5391E+00     .3489E-04     9    .7299E-03     7    .2041E-03

                  ****** ITERATION LIMIT REACHED ******
```

```
                                                FINAL OUTPUT FOR COARSE MESH

FORCE COEFFICIENTS, PRESSURE COEFFICIENT, AND MACH NUMBER
(OR SIMILARITY PARAMETER) ON BODY AND DIVIDING STREAM LINE.
                                                                                  B = BEFORE OR BEHIND AIRFOIL
       CL =    .618266                                                            U = UPPER AIRFOIL SURFACE
       CM =  -.155303                                                             L = LOWER AIRFOIL SURFACE
      CP* =  -.003170                                                             * = CRITICAL PRESSURE

                           LOWER                          UPPER
                           Y=0-                           Y=0+
  I        X          CP          M1             CP          M1

  1  -1.075000     .003000     .740905        .003000     .740905                              B                      *
  2   -.575000     .043372     .720725        .043372     .720725                            B                        *
  3   -.175000     .129825     .684349        .129825     .684349                         B                           *
  4   -.025000     .264500     .608810        .264500     .608810                    B                                *
  5   -.002500     .360006     .540448        .360006     .540448                B                                    *
                      AIRFOIL LEADING EDGE                                            AIRFOIL LEADING EDGE
  6    .017500     .630016     .324855        .249591     .617627      L              U                               *
  7    .037500     .337262     .563801       -.143732     .816590                 L                 U                 *
  8    .075000     .094363     .723235       -.062652     .985196                            L                      U *
  9    .140625    -.182317     .833511       -.760830    1.059631                                     L               *     U
 10    .265625    -.270642     .874419       -.730998    1.047965                                         L           *    U
 11    .390625    -.230724     .854314       -.739994    1.240448                                       L             *    U
 12    .515625    -.084038     .790520       -.766144    1.057455                                 L                   *     U
 13    .640625     .141340     .670215       -.652882    1.020503                         L                           *  U
 14    .765625     .364400     .546000       -.419234     .930906                L                             U      *
 15    .885000     .460487     .470009       -.110344     .805195             L                    U                  *
 16    .965000     .447856     .487498        .100112     .699918             L            U                          *
 17   1.000000     .329958     .560480        .208905     .641253                  L   U                              *
                      AIRFOIL TRAILING EDGE                                           AIRFOIL TRAILING EDGE
 18   1.090000     .147296     .675027        .147296     .675027                        B                            *
 19   1.400000     .076301     .712149        .076301     .712149                           B                         *
 20   1.875000     .029479     .735607        .029479     .735607                             B                       *


          Y-GRID VALUES     Y(J) J=   1 TO  14
       -5.200000   -2.400000   -1.150000    -.550000    -.300000    -.180000    -.075000     .075000
         .180000     .300000     .550000    1.150000    2.400000    5.200000
```

Figure 4-4.- Continued.

```
                                   INTERMEDIATE OUTPUT FOR MEDIUM MESH

        WE =   1.9000     EPS =     .2000     MAXIT FOR THIS MESH =  250

ITERATION     CL          CM      IERR  JERR      ERROR      IRL   JRL       BIGRL        ERCIRC     ICPU     CPERRU     ICPL     CPERRL

   10       .61625     -.15287     26    13    .20790E-02    13    11    .39986E+02    .12[illegible]E-02   [illegible]   .[illegible]   1[illegible]   .[illegible]
   20       .60969     -.15190      6    14    .62591E-03    13    14    .30036E+02    .12208E-02    14   .[illegible]   [illegible]   .19408E-01
   30       .60690     -.15213     19     8    .29526E-03    10    15    .82496E+01    .2461[illegible]E-03   16   .[illegible]E-02   12   .17031E-01
   40       .60546     -.15221     20    14    .20151E-03    10     6    .44496E+01    .16751E-03    17   .[illegible]   [illegible]   .7[illegible]E-02
   50       .60527     -.15230      9     7    .16391E-03    13    14    .39391E+01    .57066E-04    14   .[illegible]E-02   11   .2[illegible]E-02
   60       .60464     -.15242     23    14    .57066E-04    13    14    .18427E+01    .71[illegible]E-04   [illegible]   .[illegible]   [illegible]   .[illegible]E-03
   70       .60432     -.15242     22    14    .49460E-04    10    14    .20446E+01    .43426E-04   [illegible]   .797[illegible]   [illegible]   .1[illegible]E-03
   80       .60414     -.15243     22    14    .33966E-04    13    14    .16128E+01    .27806E-04   [illegible]   .[illegible]   10   .1[illegible]E-03
   90       .60400     -.15243     22    14    .24231E-04    13    14    .12946E+01    .2210[illegible]E-04   1[illegible]   .[illegible]   [illegible]   .[illegible]E-04
  100       .60396     -.15242     22    14    .16921E-04    10    14    .10097E+01    .15371E-04   1[illegible]   .[illegible]   [illegible]   .[illegible]E-04
  110       .60382     -.15242     22    14    .13976E-04    13    14    .78181E+00    .[illegible]E-04   1[illegible]   .22306E-03   12   .[illegible]E-04
  120       .60377     -.15242     11    14    .10638E-04    13    14    .60222E+00    .8491[illegible]E-05   1[illegible]   .[illegible]   [illegible]2   .[illegible]E-04
  130       .60373     -.15241     10    14    .81446E-05    13    14    .46161E+00    .6354[illegible]E-05   [illegible]   .1[illegible]   12   .39[illegible]E-04

                  ******  SOLUTION CONVERGED  ******
```

```
                                        FINAL OUTPUT FOR MEDIUM MESH

FORCE COEFFICIENTS, PRESSURE COEFFICIENT, AND MACH NUMBER
(OR SIMILARITY PARAMETER) ON BODY AND DIVIDING STREAM LINE.

        CL =   .603729
        CM =  -.152415
       CP* =  -.603170

                            LOWER                                UPPER
                            Y=0-                                 Y=0+
 I        X          CP          M1                      CP           M1

 1   -1.075000     .021647     .739458                .021647     .739458
 2    -.825000     .033335     .733764                .033335     .733764
 3    -.575000     .051951     .724443                .051951     .724443
 4    -.356000     .092761     .703748                .092761     .703748
 5    -.175000     .177067     .658347                .177067     .658347
 6    -.075000     .315444     .577665                .315444     .577665
 7    -.035000     .458494     .479519                .458494     .479519
 8    -.015000     .567092     .388057                .567092     .388057
 9    -.002500     .654248     .296566                .654248     .296566
                      AIRFOIL LEADING EDGE
10     .007500     .942391    0.000000                .553319     .402445
11     .017500     .592448     .364366                .172356     .661459
12     .027500     .381019     .534372               -.083733     .788[illegible]7
13     .037500     .225376     .643069               -.300637     .886527
14     .055000     .038007     .731391               -.509295     .994569
15     .075000    -.094836     .794536               -.836156    1.00120[illegible]
16     .097750    -.204463     .843093               -.907765    1.134459
17     .140625    -.294224     .880862               -.863369    1.090295
18     .203125    -.337001     .898821               -.812684    1.073290
19     .265625    -.346726     .899815               -.773472    1.066136
20     .328125    -.320384     .891645               -.750194    1.061971
21     .390625    -.276690     .874922               -.740073    1.060732
22     .453125    -.213003     .847749               -.746913    1.055884
23     .515625    -.125367     .828352               -.769921    1.059437
24     .578125    -.013309     .750447               -.797101    1.066095
25     .640625     .120502     .684277               -.756741    1.054225
26     .703125     .263412     .639463               -.597553     .995414
27     .765625     .305461     .531935               -.408063     .926543
28     .828125     .455097     .482479               -.251294     .8636[illegible]2
29     .885000     .476548     .465664               -.091152     .792855
30     .915000     .473400     .468047                .032375     .734178
31     .945000     .459460     .478787                .132221     .683184
32     .975000     .426126     .503409                .223371     .632836
33    1.000000     .345646     .558744                .282392     .598056
                      AIRFOIL TRAILING EDGE
34    1.025000     .229962     .629647                .229962     .629647
35    1.090000     .165695     .665090                .165695     .665090
36    1.225000     .110001     .694462                .110001     .694462
37    1.466600     .073623     .712494                .073623     .712494
38    1.625000     .055238     .722096                .055238     .722096
39    1.875000     .040203     .730301                .040203     .730301
```



```
           Y-GRID VALUES        Y(J) J=   1 TO  28
   -5.200000   -3.650000   -2.450000   -1.650000   -1.150000    -.800000    -.550000    -.390000
    -.300000    -.240000    -.180000    -.120000    -.075000    -.035000     .035000     .075000
     .125000     .180000     .245000     .300000     .390000     .550000     .800000    1.150000
    1.650000    2.450000    3.650000    5.200000
```

Figure 4-4.- Continued.

```
                                          INTERMEDIATE OUTPUT FOR FINE MESH

        WE =   1.9500     EPS =     .2000     MAXIT FOR THIS MESH =  500

ITERATION    CL         CM      IERR  JERR    ERROR      IRL   JRL     BIGRL        ERCIRC    ICPU    CPERRU     ICPL    CPERRL

   10      .60707   -.15260     9    29   .21046E-02   17    28   .30731E+03   .27796E-03   [illegible]   [illegible]   [illegible]   [illegible]
   20      .60871   -.15277    16    31   .81211E-03   17    28   .10208E+03   .24531E-03   34   [illegible]   21   [illegible]
   30      .61016   -.15295    18    31   .42644E-03   17    28   .13626E+03   .25389E-03   [illegible]   .25032E-01   20   .10206E-01
   40      .61131   -.15288    37    28   .34696E-03   17    28   .10291E+03   .18111E-03   [illegible]   .21806E-01   24   [illegible]
   50      .61210   -.15284    16    31   .26906E-03   17    28   .83391E+02   .11671E-03   41   .13071E-01   31   .46421E-02
   60      .61290   -.15290    18    29   .21466E-03   17    28   .70238E+02   .12796E-03   [illegible]   [illegible]   [illegible]   .24796E-02
   70      .61353   -.15288    18    29   .16946E-03   17    28   .59681E+02   .11276E-03   44   .80498E-02   24   [illegible]
   80      .61424   -.15294    18    29   .15066E-03   17    28   .39981E+02   .11438E-03   44   .72162E-02   24   .14822E-02
   90      .61486   -.15293    18    29   .13451E-03   17    28   .43831E+02   .85721E-04   42   .60896E-02   24   .12091E-02
  100      .61527   -.15293    18    29   .11606E-03   17    28   .37696E+02   .77371E-04   42   .62626E-02   24   .16061E-02
  110      .61568   -.15294    19    29   .10326E-03   17    28   .32486E+02   .64438E-04   42   .51774E-02   24   .86906E-03
  120      .61601   -.15293    18    29   .87386E-04   17    28   .28296E+02   .54246E-04   44   .47721E-02   24   .71296E-03
  130      .61631   -.15294    18    29   .75211E-04   17    28   .24481E+02   .48966E-04   42   .39466E-02   24   .58916E-03
  140      .61658   -.15294    18    29   .65406E-04   17    28   .21311E+02   .44671E-04   42   .32096E-02   24   .34671E-03
  150      .61682   -.15294    18    29   .56996E-04   17    28   .18506E+02   .38811E-04   42   [illegible]   24   [illegible]
  160      .61703   -.15295    19    29   .49986E-04   17    28   .16066E+02   .33446E-04   42   .22396E-02   24   .35596E-03
  170      .61721   -.15295    18    29   .42086E-04   17    28   .13986E+02   .28136E-04   42   .21626E-02   24   .34316E-03
  180      .61736   -.15295    18    29   .37626E-04   17    28   .12106E+02   .25226E-04   42   [illegible]   24   [illegible]
  190      .61750   -.15295    18    29   .32616E-04   17    28   .10621E+02   .22106E-04   44   .13963E-02   24   .25662E-03
  200      .61762   -.15295    18    29   .28426E-04   17    28   .92586E+01   .19556E-04   42   .13076E-02   24   .23946E-03
  210      .61773   -.15295    18    29   .24786E-04   17    28   .80721E+01   .17248E-04   42   [illegible]   24   [illegible]
  220      .61782   -.15295    18    29   .21616E-04   17    28   .79408E+01   .14951E-04   42   .12206E-02   24   .16916E-03
  230      .61790   -.15295    18    29   .18856E-04   17    28   .61426E+01   [illegible]   42   [illegible]   24   .14296E-03
  240      .61797   -.15295    18    29   .16446E-04   17    28   .53596E+01   .11351E-04   42   .74771E-03   24   .12468E-03
  250      .61803   -.15295    18    29   .14356E-04   17    28   .46776E+01   .99186E-05   42   .64206E-03   24   .10616E-03
  260      .61808   -.15295    18    29   .12536E-04   17    28   .40826E+01   .86746E-05   42   .61296E-03   24   .94296E-04
  270      .61813   -.15295    18    29   .10931E-04   17    28   .35626E+01   .75786E-05   42   .82296E-03   24   .82316E-04
  280      .61817   -.15295    18    29   .95386E-05   17    28   .31796E+01   .66116E-05   42   .49916E-03   24   .72806E-04

                    ******  SOLUTION CONVERGED  ******




                                                                   ******************
                                                                   *                *
                                                                   *  FINAL OUTPUT  *
                                                                   *                *
                                                                   ******************

PRINTOUT IN  PHYSICAL  VARIABLES.

DEFINITION OF SIMILARITY PARAMETERS BY KRUPP

BOUNDARY CONDITION FOR FREE AIR

DIFFERENCE EQUATIONS ARE FULLY CONSERVATIVE.

KUTTA CONDITION IS ENFORCED.

             MACH =     .7500000
            DELTA =     .1150000
            ALPHA =     .1200000
                K =    2.4087108
 DOUBLET STRENGTH =     .0715562
   AIRFOIL VOLUME =     .0755062

        PARAMETERS USED TO TRANSFORM VARIABLES
                TO TRANSONIC SCALING

           CPFACT =     .2934287
           CDFACT =     .0337443
           CMFACT =     .2934287
           CLFACT =     .2934287
            YFACT =    2.3744886
            VFACT =    6.5096146
```

Figure 4-4.- Continued.

```
                                              FINAL OUTPUT FOR FINE MESH

FORCE COEFFICIENTS, PRESSURE COEFFICIENT, AND MACH NUMBER
(OR SIMILARITY PARAMETER) ON BODY AND DIVIDING STREAM LINE.
                                                                                   B = BEFORE OR BEHIND AIRFOIL
        CL =   .618170                                                             U = UPPER AIRFOIL SURFACE
        CM =  -.192947                                                             L = LOWER AIRFOIL SURFACE
       CP* =  -.603170                                                             * = CRITICAL PRESSURE

                               LOWER                          UPPER
                               Y=0-                           Y=0+
 I        X          CP          M1              CP          M1

 1   -1.075000    .034632     .733359         .034632     .733359
 2    -.950000    .038228     .731281         .038228     .731281
 3    -.825000    .043603     .728611         .043603     .728611
 4    -.700000    .051881     .724476         .051881     .724476
 5    -.575000    .064692     .718036         .064692     .718036
 6    -.450000    .084149     .708141         .084149     .708141
 7    -.350000    .112631     .693421         .112631     .693421
 8    -.250000    .156104     .673278         .156104     .673278
 9    -.175000    .217644     .6364520        .217644     .636452
10    -.125000    .297629     .588744         .297629     .588744
11    -.075000    .410957     .514216         .410957     .514216
12    -.052500    .536533     .416334         .536533     .416334
13    -.035000    .662740     .285993         .662740     .285993
14    -.022500    .794714    0.000000         .794714    0.000000
15    -.015000    .918871    0.000000         .918871    0.000000
16    -.007500   1.041777    0.000000        1.041777    0.000000
17    -.002500   1.163438    0.000000        1.163438    0.000000
                      AIRFOIL LEADING EDGE
18     .002500   1.407124    0.000000         .929811    0.000000
19     .007500    .847589    0.000000         .304417     .546055
20     .012500    .545956     .438849         .069857     .715423
21     .017500    .358894     .549707        -.133933     .812166
22     .022500    .224692     .629549        -.262543     .876436
23     .027500    .134061     .682093        -.393786     .923939
24     .032500    .061736     .719528        -.483822     .954598
25     .037500   -.021223     .755991        -.567411     .986946
26     .045000   -.069930     .781231        -.646691    1.015661
27     .055000   -.126525     .808871        -.741993    1.049139
28     .065000   -.175851     .830693        -.849193    1.083564
29     .075000   -.215274     .847733        -.938452    1.114969
30     .085000   -.248385     .861778       -1.063396    1.136551
31     .097500   -.281602     .875431       -1.023934    1.124245
32     .115000   -.315831     .889710       -1.008197    1.137445
33     .140625   -.346298     .902143        -.979322    1.128201
34     .171875   -.364996     .909533        -.949933    1.118713
35     .203125   -.371656     .912179        -.923959    1.110261
36     .234375   -.371149     .911980        -.903749    1.102653
37     .265625   -.365010     .909857        -.879652    1.095487
38     .296875   -.355271     .905644        -.859339    1.088532
39     .328125   -.340392     .899666        -.836023    1.081159
40     .359375   -.320585     .891637        -.811060    1.072749
41     .390625   -.295284     .881294        -.781224    1.062614
42     .421875   -.264496     .868532        -.743156    1.049541
43     .453125   -.227512     .852449        -.701266    1.034965
44     .484375   -.183297     .834790        -.682262    1.028204
45     .515625   -.130117     .814052        -.709964    1.034839
46     .546875   -.084262     .789704        -.763614    1.048455
47     .578125   -.023989     .763512        -.782223    1.062955
48     .609375    .042142     .729337        -.815561    1.075282
49     .640625    .114148     .693160        -.836069    1.081472
50     .671875    .187714     .652951        -.772353    1.059967
51     .703125    .263167     .609439        -.591992     .995937
52     .734375    .333402     .566279        -.467771     .949627
53     .765625    .392670     .522951        -.417877     .930377
54     .796875    .436765     .495695        -.342758     .903614
55     .828125    .464615     .474867        -.257138     .855454
56     .859375    .476299     .464249        -.166233     .820483
57     .885000    .482335     .461135        -.088461     .793704
58     .900000    .481972     .461421        -.026327     .762624
59     .915000    .479614     .463270         .025433     .737599
60     .930000    .475291     .466639         .076493     .711797
61     .945000    .468316     .472032         .128491     .685655
62     .960000    .457559     .481220         .183440     .658978
63     .975000    .438967     .494111         .232955     .627820
64     .990000    .407746     .516974         .282117     .598223
65    1.000000    .353399     .553324         .314941     .577661
                      AIRFOIL TRAILING EDGE
66    1.010000    .285581     .596123         .285582     .596120
67    1.025000    .248178     .618456         .248178     .618456
68    1.050000    .209303     .640647         .209303     .640647
69    1.090000    .170951     .662196         .170951     .662196
70    1.150000    .136726     .680682         .136726     .680682
71    1.225000    .110219     .694661         .110219     .694661
72    1.300000    .090361     .704952         .090361     .704952
73    1.400000    .074751     .712938         .074751     .712938
74    1.500000    .062765     .719039         .062[illegible]     .719039
75    1.625000    .053453     .723691         .05[illegible]     .723691
76    1.750000    .046808     .727062         .04[illegible]     .727062
77    1.875000    .041477     .729668         .041477     .729668
```

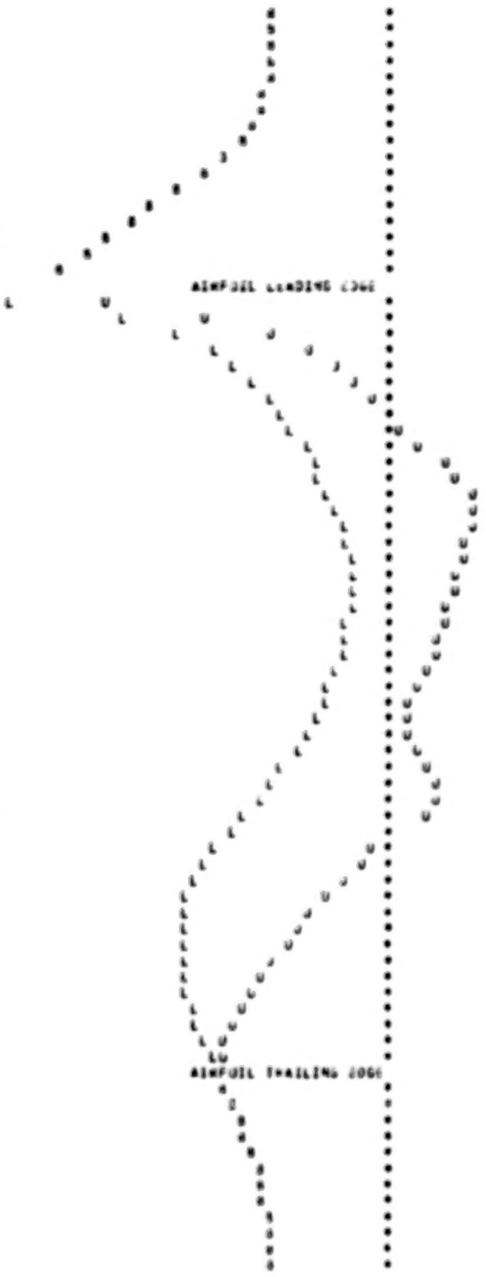


Figure 4-4.- Continued.

```
   Y-GRID VALUES      Y(J) J=   1 TO  56
 -5.200000   -4.400000   -3.600000   -3.000000   -2.400000   -1.950000   -1.600000   -1.350000
 -1.190000    -.950000    -.800000    -.650000    -.550000    -.450000    -.390000    -.345000
  -.300000    -.270000    -.240000    -.210000    -.180000    -.150000    -.125000    -.100000
  -.075000    -.050000    -.030000    -.010000     .010000     .030000     .050000     .075000
   .100000     .125000     .150000     .180000     .210000     .240000     .270000     .300000
   .345000     .390000     .450000     .550000     .650000     .800000     .950000    1.190000
  1.350000    1.600000    1.950000    2.400000    3.000000    3.600000    4.400000    5.200000


                                      FLOW CHARACTER MAP
                                      ******************

                P = PARABOLIC
                H = HYPERBOLIC
                S = SHOCK
                - = ELLIPTIC

 56     ------------------------------------------------------------------
 55     ------------------------------------------------------------------
 54     ------------------------------------------------------------------
 53     ------------------------------------------------------------------
 52     ------------------------------------------------------------------
 51     ------------------------------------------------------------------
 50     ------------------------------------------------------------------
 49     ------------------------------------------------------------------
 48     ------------------------------------------------------------------
 47     ------------------------------------------------------------------
 46     ------------------------------------------------------------------
 45     ------------------------------------------------------------------
 44     ------------------------------------------------------------------
 43     ------------------------------------------------------------------
 42     ------------------------------------------------------------------
 41     ------------------------------------------------------------------
 40     ------------------------------PHHHS-------------------------------
 39     ------------------------------PHHHHS------------------------------
 38     -----------------------------PHHHHHS------------------------------
 37     ----------------------------PHHHHHHHS-----------------------------
 36     ----------------------------PHHHHHHHS--PHHS-----------------------
 35     ---------------------------PHHHHHHHHHSPHHHS-----------------------
 34     ---------------------------PHHHHHHHHHHHHHHS-----------------------
 33     ---------------------------PHHHHHHHHHHHHHHHS----------------------
 32     --------------------------PHHHHHHHHHHHHHHHHS----------------------
 31     ------------------------PHHHHHHHHHHHHHHHHHHS----------------------
 30     -----------------------PHHHHHHHHHHHHHHHHHHHS----------------------
 29     ----------------------PHHHHHHHHHHHHHHHHHHHHS----------------------
 28     ------------------------------------------------------------------
 27     ------------------------------------------------------------------
 26     ------------------------------------------------------------------
 25     ------------------------------------------------------------------
 24     ------------------------------------------------------------------
 23     ------------------------------------------------------------------
 22     ------------------------------------------------------------------
 21     ------------------------------------------------------------------
 20     ------------------------------------------------------------------
 19     ------------------------------------------------------------------
 18     ------------------------------------------------------------------
 17     ------------------------------------------------------------------
 16     ------------------------------------------------------------------
 15     ------------------------------------------------------------------
 14     ------------------------------------------------------------------
 13     ------------------------------------------------------------------
 12     ------------------------------------------------------------------
 11     ------------------------------------------------------------------
 10     ------------------------------------------------------------------
  9     ------------------------------------------------------------------
  8     ------------------------------------------------------------------
  7     ------------------------------------------------------------------
  6     ------------------------------------------------------------------
  5     ------------------------------------------------------------------
  4     ------------------------------------------------------------------
  3     ------------------------------------------------------------------
  2     ------------------------------------------------------------------
  1     ------------------------------------------------------------------
```

Figure 4-4.- Continued.

```
                                              MACH NUMBER MAP
                                              ***************

SYMBOL       RANGE               SYMBOL       RANGE               SYMBOL       RANGE               SYMBOL       RANGE
  0             M.LE.0.05          A   0.95.LT.M.LE.1.05          K   1.95.LT.M.LE.2.05          U   2.95.LT.M.LE.3.05
  1   0.05.LT.M.LE.0.15          B   1.05.LT.M.LE.1.15          L   2.05.LT.M.LE.2.15          V   3.05.LT.M.LE.3.15
  2   0.15.LT.M.LE.0.25          C   1.15.LT.M.LE.1.25          M   2.15.LT.M.LE.2.25          W   3.15.LT.M.LE.3.25
  3   0.25.LT.M.LE.0.35          D   1.25.LT.M.LE.1.35          N   2.25.LT.M.LE.2.35          X   3.25.LT.M.LE.3.35
  4   0.35.LT.M.LE.0.45          E   1.35.LT.M.LE.1.45          O   2.35.LT.M.LE.2.45          Y   3.35.LT.M.LE.3.45
  5   0.45.LT.M.LE.0.55          F   1.45.LT.M.LE.1.55          P   2.45.LT.M.LE.2.55          Z   3.45.LT.M.LE.3.55
  6   0.55.LT.M.LE.0.65          G   1.55.LT.M.LE.1.65          Q   2.55.LT.M.LE.2.65          *             M.GT.3.55
  7   0.65.LT.M.LE.0.75          H   1.65.LT.M.LE.1.75          R   2.65.LT.M.LE.2.75
  8   0.75.LT.M.LE.0.85          I   1.75.LT.M.LE.1.85          S   2.75.LT.M.LE.2.85
  9   0.85.LT.M.LE.0.95          J   1.85.LT.M.LE.1.95          T   2.85.LT.M.LE.2.95
```

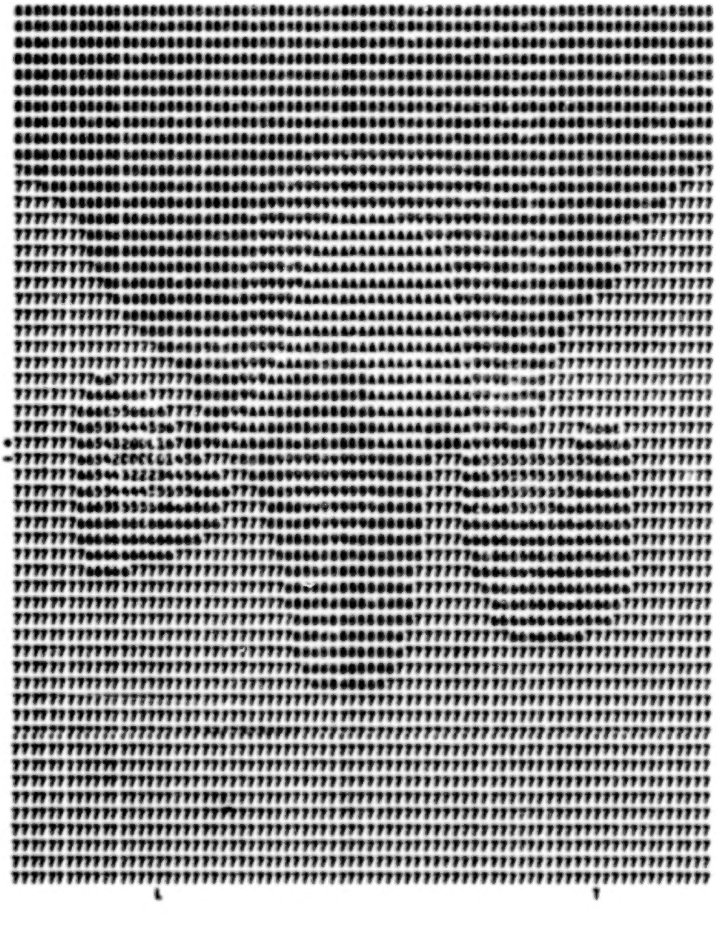
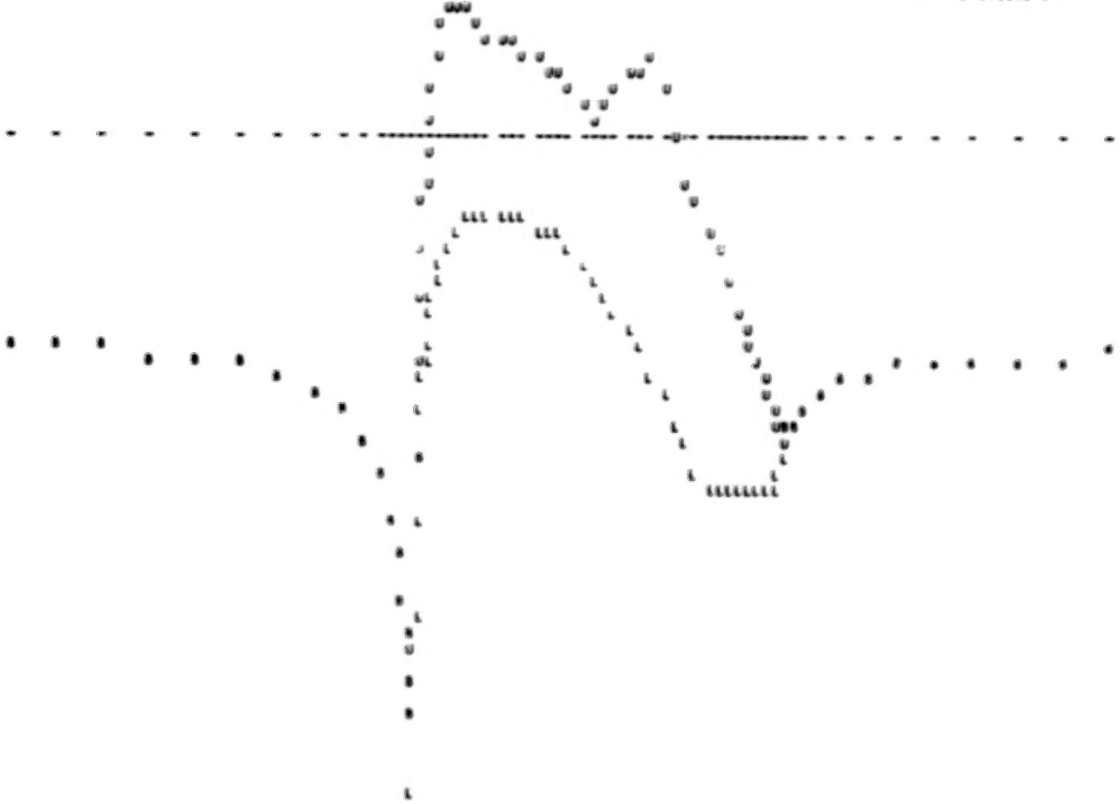


Figure 4-4.- Continued.

PRINTER PLOT OF CP ON BODY AND DIVIDING STREAMLINE
****

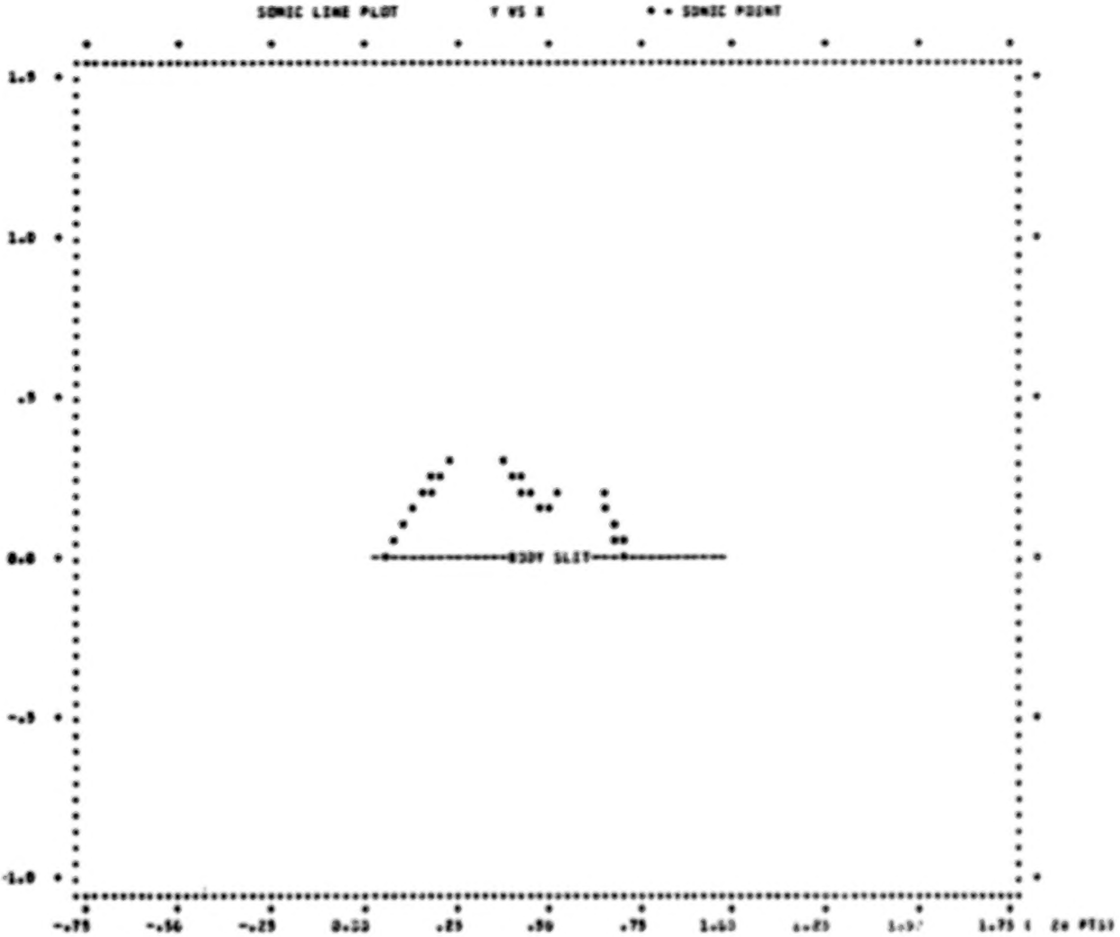
U = CP(UPPER)
L = CP(LOWER)
B = CP(UPPER) = CP(LOWER)
--- = CRITICAL CP



```
                                            SONIC LINE OUTPUT
                                            *****************
SONIC LINE COORDINATES

    Y           X-SONIC

  .30000     .23420     .37126
  .27000     .20723     .39561
  .24000     .18509     .41554
  .21000     .16501     .43445
  .18000     .14349     .45086     .52727     .63785
  .15000     .12669     .48161     .48769     .65392
  .12500     .11104     .66557
  .10000     .09882     .67579
  .07500     .08642     .68326
  .05000     .07393     .68982
  .03000     .06207     .69457
  .01000     .04913     .69902
 BODY LOCATION
```

Figure 4-4.- Continued.

```
                         SONIC LINE OUTPUT
                         *****************
```



```
                  SHOCK WAVE DRAG AND TOTAL PRESSURE PROFILE OUTPUT
                  *************************************************

INVISCID WAKE PROFILES FOR INDIVIDUAL SHOCK WAVES WITHIN MOMENTUM CONTOUR

SHOCK  1
WAVE DRAG FOR THIS SHOCK=     .000149
     Y            CD(Y)        PO/POINF

  0.00000000   .00141597    .99937810
   .01000000   .00138824    .99939560
   .03000000   .00133524    .99913025
   .05000000   .00123766    .99919380
   .07500000   .00107696    .99929868
   .10000000   .00089976    .99941307
   .12500000   .00053393    .99976246
   .15000000   .00030774    .99982412
   .18000000   .00003761    .99997550
```

Figure 4-4.- Continued.

```
                                                              DRAG COEFFICIENT OUTPUT
                                                              ***********************

CALCULATION OF DRAG COEFFICIENT BY MOMENTUM INTEGRAL METHOD

BOUNDARIES OF CONTOUR USED                  CONTRIBUTION TO CD
UPSTREAM    X =     -.179000                 CDUP    =    -.002898
DOWNSTREAM  X =     1.229000                 CDDOWN  =     .003679
TOP         Y =     4.400000                 CDTOP   =     .000624
BOTTOM      Y =    -4.400000                 CDBOT   =    -.002000
               TOTAL CONTRIBUTIONS AROUND CONTOUR =    -.003283

THERE ARE  1 SHOCKS INSIDE CONTOUR. TOTAL CDWAVE =     .003149

NOTE - ALL SHOCKS CONTAINED WITHIN CONTOUR
CDWAVE EQUALS TOTAL WAVE DRAG

DRAG CALCULATED FROM MOMENTUM INTEGRAL    CD     =    -.000094



  TIME TO RUN CASE WAS  16.49 SECONDS.
```

(b) Output.

Figure 4-4.- Concluded

```
SAMPLE CASE 2 - SUBSONIC-NACA0006 AIRFOIL-SOLID TUNNEL WALL SIMULATION
$INP
 BCTYPE=2, H=2.0, ALPHA=2.0, BCFOIL=1, DELTA=0.06, EMACH=0.75,
$END
```

(a) Input.

```
SAMPLE CASE 2 - SUBSONIC - NACA0006 AIRFOIL - SOLID TUNNEL WALL SIMULATION


                                        INPUT PARAMETERS
                                        ****************

    EMACH =   .75000       POR =  0.00000      IMIN =    1     BCTYPE =  2        AMESH = F

    DELTA =   .06000     CLSET =  0.00000     IMAXI =   77     BCFOIL =  1         PHYS = T

    ALPHA =  2.00000       EPS =   .20000      JMIN =    1     PSTART =  1        PSAVE = F

       AK =  0.00000      RIGF =  0.00000     JMAXI =   48     PRTFLO =  1        KUTTA = T

      GAM =  1.40000     WCIRC =  1.00000     MAXIT =  500     IPRTER = 10          FCR = T

        F =  0.00000    CVERGE =   .00001        NU =  100     SIMDEF =  3

        H =  2.00000    DVERGE =     10.0        NL =   75       ICUT =  2

       WE = 1.80,1.90,1.95

     XIN
    -1.075000     -.950000     -.825000     -.700000     -.575000     -.450000     -.350000     -.250000
     -.175000     -.125000     -.075000     -.052500     -.037500     -.022500     -.012500     -.007500
     -.002500      .002500      .007500      .012500      .017500      .022500      .027500      .032500
      .037500      .045000      .055000      .065000      .075000      .085000      .097500      .115000
      .140625      .171875      .203125      .234375      .265625      .296875      .328125      .359375
      .390625      .421875      .453125      .484375      .515625      .546875      .578125      .609375
      .640625      .671875      .703125      .734375      .765625      .796875      .828125      .859375
      .885000      .900000      .915000      .930000      .945000      .960000      .975000      .990000
     1.000000     1.012500     1.025000     1.050000     1.100000     1.175000     1.250000     1.325000
     1.400000     1.500000     1.625000     1.750000     1.875000

     YIN
    -2.000000    -1.862075    -1.663468    -1.404406    -1.208383    -1.000000     -.843346     -.697216
     -.550000     -.455000     -.390000     -.340000     -.300000     -.270000     -.246000     -.220000
     -.190000     -.150000     -.125000     -.100000     -.075000     -.050000     -.030000     -.010000
      .010000      .030000      .050000      .075000      .100000      .125000      .150000      .190000
      .220000      .246000      .270000      .300000      .340000      .390000      .455000      .550000
      .697216      .843346     1.000000     1.208383     1.404406     1.663468     1.862075     2.000000



                                        AIRFOIL GEOMETRY OUTPUT
                                        ***********************

PRINTOUT IN PHYSICAL VARIABLES NORMALIZED BY CHORD LENGTH                             MAX THICKNESS =   .03001529
AIRFOIL VOLUME=    .04100963                                                          MAX CAMBER    =  0.00000000

 INPUT  GRID          UPPER   SURFACE                    LOWER   SURFACE
     X                  Y            DY/DX                  Y            DY/DX          THICKNESS     CAMBER

  .00250000         .00435834     .85237420          -.00435834    -.85237420         .00435834   0.00000000
  .00750000         .00742429     .47487803          -.00742429    -.47487803         .00742429   0.00000000
  .01250000         .00946951     .35793589          -.00946951    -.35793589         .00946951   0.00000000
  .01750000         .01108950     .29523886          -.01108950    -.29523886         .01108950   0.00000000
  .02250000         .01245756     .25448155          -.01245756    -.25448155         .01245756   0.00000000
  .02750000         .01365308     .22514572          -.01365308    -.22514572         .01365308   0.00000000
  .03250000         .01472030     .20264561          -.01472030    -.20264561         .01472030   0.00000000
  .03750000         .01568694     .18462015          -.01568694    -.18462015         .01568694   0.00000000
  .04500000         .01698765     .16315384          -.01698765    -.16315384         .01698765   0.00000000
  .05500000         .01850460     .14124882          -.01850460    -.14124882         .01850460   0.00000000
  .06500000         .01982871     .12421584          -.01982871    -.12421584         .01982871   0.00000000
  .07500000         .02099952     .11038471          -.02099952    -.11038471         .02099952   0.00000000
  .08500000         .02204384     .09879605          -.02204384    -.09879605         .02204384   0.00000000
  .09750000         .02320018     .08657702          -.02320018    -.08657702         .02320018   0.00000000
  .11500000         .02458752     .07246479          -.02458752    -.07246479         .02458752   0.00000000
  .14062500         .02622499     .05601496          -.02622499    -.05601496         .02622499   0.00000000
  .17187500         .02772009     .04030395          -.02772009    -.04030395         .02772009   0.00000000
```

Figure 4-5.- Sample test case 2 - subsonic, solid wall wind-tunnel simulation (full output): NACA 0006 airfoil, δ = 0.06, α = 2°, M_∞ = 0.75, H = 2.

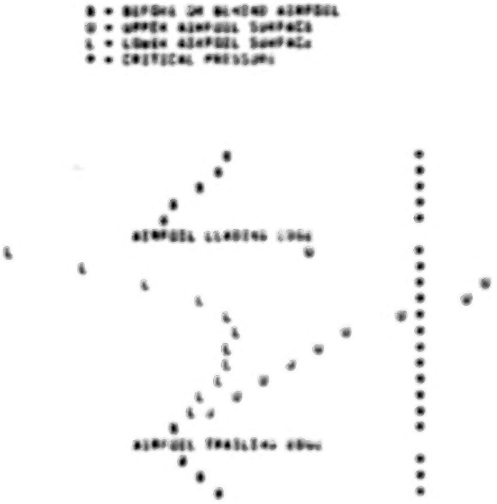
Microfilmed From
Best Available Copy

```
[illegible]
```

```
                         INTERMEDIATE OUTPUT FOR COARSE MESH

      WE =   1.0000    EPS =   .2000    MAXIT FOR THIS MESH =  125

ITERATION    CL          CM       IERR  JERR     ERROR      IRL   JRL     BIGRL        DACIRC      ICPU  [illegible]  ICPL  [illegible]

    10     .15369   [illegible]
    20     .25112   [illegible]
    30     .29076   [illegible]
    40     .31005   [illegible]
    50     .34073   [illegible]
    60     .36199   [illegible]
    70     .37309   [illegible]
    80     .38526   [illegible]
    90     .39250   [illegible]
   100     .39797   [illegible]
   110     .40201   [illegible]
   120     .40501   [illegible]

              ****** ITERATION LIMIT REACHED ******
```

Figure 4.5.- Continued

```
                                   FINAL OUTPUT FOR COARSE MESH

FORCE COEFFICIENTS, PRESSURE COEFFICIENT, AND MACH NUMBER
(OR SIMILARITY PARAMETER) ON BODY AND DIVIDING STREAM LINE.

        CL =    .406215
        CM =   -.003640
       CP* =   -.683175

                            LOWER                              UPPER
                             Y=0-                               Y=0+
 I         X           CP          M1                  CP          M1

 1   -1.075000    -.010671     .750440            -.010671     .750440
 2    -.575000     .003335     .746209             .003335     .746209
 3    -.175000     .033214     .723011             .033214     .723011
 4    -.019063     .120443     .686129             .120443     .686129
 5    -.002500     .167177     .654250             .167177     .654250
                      AIRFOIL LEADING EDGE
 6     .017500     .422000     .332997            -.201630     .875646
 7     .037500     .414301     .511053            -.993233     .995297
 8     .079000     .219512     .625030            -.752709    1.051121
 9     .140625     .094571     .723135            -.721066    1.041007
10     .265625    -.013574     .766661            -.534411     .974743
11     .390625    -.049098     .773752            -.396432     .919014
12     .515625    -.034713     .768965            -.293284     .882471
13     .640625    -.019112     .759376            -.208677     .844903
14     .765625     .000226     .746612            -.127227     .824185
15     .885000     .045206     .727012            -.045289     .771587
16     .945000     .086491     .706030             .033511     .725098
17    1.000000     .147961     .674659             .129151     .684611
                      AIRFOIL TRAILING EDGE
18    1.090000     .101100     .699402             .101120     .699402
19    1.460000     .043505     .728659             .043505     .728659
20    1.875000     .012993     .743691             .012993     .743691
```



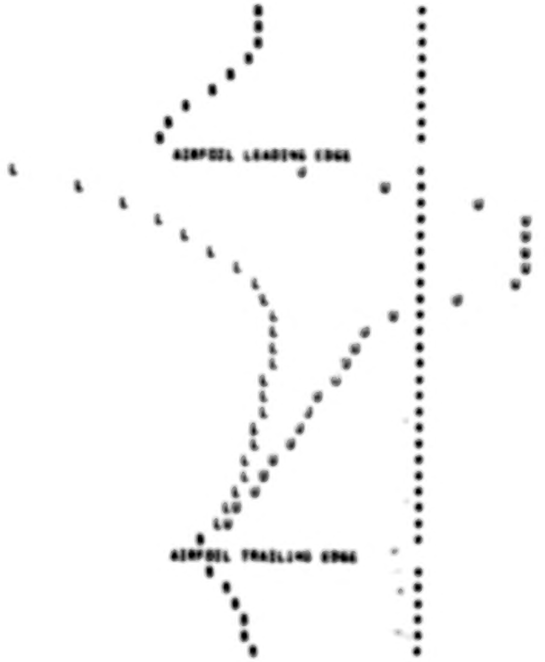
```
          Y-GRID VALUES      Y(J) J=   1 TO  12
   -2.000000   -1.284580    -.550067    -.202320    -.106230    -.079553     .075000     .145303
     .500000     .550000    1.200000    2.000000



                                 INTERMEDIATE OUTPUT FOR MEDIUM MESH

       WE =   1.9000     EPS =    .2000     MAXIT FOR THIS MESH =  250

ITERATION     CL          CM       IERR  JERR      ERROR      IRL   JRL       BIGRL          ERCIRC      ICPU      CPERRU       ICPL       CPERRL

    10      .40966      .00109      9    10     .23956-02     9    12     .11961+03     .14621-10   [illegible] [illegible] [illegible] [illegible]
    20      .40958      .10083     16    13     .29564-02    10    12     .76458+02     .46238-11   [illegible] [illegible] [illegible] [illegible]
    30      .41073      .00080     16    13     .33436-02    12    12     .37707+02     .49728-09   [illegible] [illegible] [illegible] [illegible]
    40      .41180      .00091     16    13     .36655-02    12    12     .56480+02     .59561-10   [illegible] [illegible] [illegible] [illegible]
    50      .41275      .00093     16    13     .40005-02    13    12     .41281+02     .71671-10   [illegible] [illegible] [illegible] [illegible]
    60      .41357      .00093     16    13     .42711-02    13    12     .34601+02     .81251-09   [illegible] [illegible] [illegible] [illegible]
    70      .41433      .00093     16    13     .45031-02    13    12     .30756+02     .89611-10   [illegible] [illegible] [illegible] [illegible]
    80      .41506      .00096     17    13     .47116-02    13    12     .26551+02     .97096-10   [illegible] [illegible] [illegible] [illegible]
    90      .41558      .00097     17    13     .49100-02    10    12     .25150+02     .10516-09   [illegible] [illegible] [illegible] [illegible]
   100      .41610      .00098     17    13     .50916-02    14    12     .25915+02     .11196-09   [illegible] [illegible] [illegible] [illegible]
   110      .41655      .00098     17    13     .52350-02    14    13     .21290+02     .11696-09   [illegible] [illegible] [illegible] [illegible]
   120      .41695      .00099     17    13     .53625-02    14    13     .21611+02     .12166-09   [illegible] [illegible] [illegible] [illegible]
   130      .41724      .00099     17    13     .54716-02    14    13     .21098+02     .12616-09   [illegible] [illegible] [illegible] [illegible]
   140      .41759      .00100     17    13     .55605-02    14    13     .22145+02     .12901-09   [illegible] [illegible] [illegible] [illegible]
   150      .41783      .00100     17    13     .56505-02    14    13     .22756+02     .13061-09   [illegible] [illegible] [illegible] [illegible]
   160      .41808      .00100     17    13     .57101-02    14    13     .22571+02     .13396-09   [illegible] [illegible] [illegible] [illegible]
   170      .41828      .00100     17    13     .57606-02    14    13     .22660+02     .13826-09   [illegible] [illegible] [illegible] [illegible]
   180      .41845      .00101     17    13     .58061-02    14    13     .22820+02     .14056-09   [illegible] [illegible] [illegible] [illegible]
   190      .41860      .00101     17    13     .58455-02    14    13     .22956+02     .14741-09   [illegible] [illegible] [illegible] [illegible]
   200      .41873      .00101     17    13     .59260-02    14    13     .23055+02     .14411-09   [illegible] [illegible] [illegible] [illegible]
   210      .41885      .00101     17    13     .59961-02    14    13     .23135+02     .14541-09   [illegible] [illegible] [illegible] [illegible]
   220      .41895      .00101     17    13     .59601-02    14    13     .23216+02     .14661-09   [illegible] [illegible] [illegible] [illegible]
   230      .41904      .00101     17    13     .60191-02    14    13     .23201+02     .14781-09   [illegible] [illegible] [illegible] [illegible]
   240      .41911      .00101     17    13     .60445-02    14    13     .23351+02     .14881-09   [illegible] [illegible] [illegible] [illegible]
   250      .41917      .00101     17    13     .60645-02    14    13     .23406+02     .14961-09   [illegible] [illegible] [illegible] [illegible]

                   ******  ITERATION LIMIT REACHED  ******
```

Figure 4.5.- Continued

Microfilmed From
Best Available Copy

```
                                              FINAL OUTPUT FOR MEDIUM MESH

FORCE COEFFICIENTS, PRESSURE COEFFICIENT, AND MACH NUMBER
(OR SIMILARITY PARAMETER) ON BODY AND DIVIDING STREAM LINE.

         CL =    .428179
         CM =    .000813
        CP* =  -.603170

                          LOWER                          UPPER
                          Y=0-                           Y=0+
  I       X          CP          M1              CP          M1

  1   -1.075000    .002005     .749838         .002003     .749838
  2    -.625000    .007675     .746573         .007670     .746573
  3    -.375000    .016663     .741099         .016663     .741099
  4    -.330000    .049403     .738162         .049403     .738162
  5    -.175000    .093129     .703922         .093139     .703922
  6    -.075000    .182941     .656149         .181941     .656149
  7    -.035000    .273022     .605231         .273022     .605231
  8    -.015000    .330124     .563247         .330124     .563247
  9    -.002500    .376040     .550079         .376040     .550079
                        AIRFOIL LEADING EDGE
 10     .007500    .910290    0.000000       -.155460     .820890
 11     .017500    .667224     .200149       -.602750     .955530
 12     .027500    .500125     .448271       -.707780    1.060251
 13     .037500    .374681     .537719       -.900251    1.120499
 14     .050000    .255099     .615092       -.965940    1.128126
 15     .075000    .190313     .669809       -.966560    1.124029
 16     .097500    .050143     .718149       -.969540    1.123748
 17     .140625    .012702     .744769       -.952109    1.112946
 18     .203125   -.034600     .766984       -.796319    1.094800
 19     .265625   -.055525     .776294       -.518140     .968677
 20     .328125   -.061032     .774333       -.419564     .929907
 21     .390625   -.060337     .778752       -.361770     .916197
 22     .453125   -.054600     .779900       -.356630     .898088
 23     .515625   -.063946     .771899       -.291630     .879791
 24     .578125   -.059562     .766090       -.248523     .861037
 25     .640625   -.023147     .761111       -.207256     .844284
 26     .703125   -.009215     .754443       -.165900     .826840
 27     .765625    .007169     .748523       -.122602     .807573
 28     .828125    .027783     .736402       -.077675     .766573
 29     .885000    .050570     .725130       -.030166     .766448
 30     .915000    .074440     .713890        .000169     .754039
 31     .945000    .100690     .699625        .051955     .724441
 32     .975000    .161700     .677979        .100949     .695272
 33    1.000000    .199629     .640411        .192314     .658891
                        AIRFOIL TRAILING EDGE
 34    1.025000    .153756     .671947         .153756     .671947
 35    1.090000    .092927     .703651         .092927     .703651
 36    1.225000    .052000     .724419         .052000     .724419
 37    1.400000    .030676     .739017         .030676     .739017
 38    1.625000    .019667     .740430         .019667     .740430
 39    1.875000    .011119     .744404         .011119     .744404
```

B = BEFORE OR BEHIND AIRFOIL
U = UPPER AIRFOIL SURFACE
L = LOWER AIRFOIL SURFACE
* = CRITICAL PRESSURE



```
         Y-GRID VALUES      Y(J) J=   1 TO  24
   -2.000000   -1.600000   -1.250000    -.600000    -.350000    -.290000    -.160000    -.240000
    -.100000    -.125000    -.075000    -.030000     .030000     .075000     .125000     .100000
     .240000     .300000     .390000     .790000     .600000    1.250000    1.600000    2.000000


                                       INTERMEDIATE OUTPUT FOR FINE MESH

        WE =   1.9500     EPS =    .2000     MAXIT FOR THIS MESH =  500

ITERATION    CL        CM     IERR  JERR    ERROR     IRL   JRL     BIGRL       ERCIRC    ICPU    CPERRU    ICPL    CPERRL

    10     .42002    .00326     9    25  .23610-02   17    24  .40377+03  .81661-04   33  .17710+01   18  .17701+01
    20     .42096    .00327    17    17  .90646-03   17    24  .20360+03  .14310-03   36  .90440-01   23  .17300-01
    30     .42107    .00340    72    36  .40960-03   17    24  .19750+03  .13550-03   36  .32320-01   19  .79000-02
    40     .42159    .00343    16    20  .34290-03   17    24  .14575+03  .13071-03   36  .26750-01   21  .56620-02
    50     .42210    .00346    16    27  .29030-03   17    24  .11330+03  .13460-03   36  .13660-01   23  .26690-02
    60     .42250    .00342    16    20  .21930-03   17    24  .96330+02  .12300-03   36  .16090-01   22  .14620-02
    70     .42302    .00343    16    27  .19900-03   17    24  .82600+02  .11400-03   36  .11700-01   23  .10520-02
    80     .42343    .00342    21     1  .17730-03   17    24  .74360+02  .10790-03   36  .83300-02   24  .12660-02
    90     .42383    .00341    18     1  .19900-03   17    24  .65670+02  .10020-03   36  .63720-02   21  .63001-03
   100     .42419    .00342    12     1  .14160-03   17    24  .59670+02  .92750-04   36  .66390-02   28  .25720-03
   110     .42455    .00342    16     4  .12790-03   17    24  .55270+02  .87771-04   36  .52010-02   24  .20630-03
   120     .42485    .00342    14     4  .11710-03   17    24  .51900+02  .83790-04   36  .49030-02   57  .27040-03
   130     .42516    .00343    13     6  .10030-03   17    24  .47610+02  .79010-04   36  .46200-02   24  .19830-03
   140     .42543    .00343    18     1  .10701-03   17    24  .43971+02  .74601-04   36  .41030-02   62  .23630-03
   150     .42572    .00344    18     3  .93300-04   17    24  .40690+02  .70071-04   36  .42030-02   52  .11900-03
   160     .42593    .00344    16     2  .88501-04   17    24  .37080+02  .65041-04   36  .30100-02   24  .90000-04
```

Figure 4.5.- Continued

```
170     .42627     .00345    14     1    .8208E-04    17    24    .3580E+02    .6218E-04    35    .3572E-02    57    .1552E-03
180     .42645     .00345    15     1    .7606E-04    17    24    .3257E+02    .5819E-04    36    .3298E-02    24    .1896E-03
190     .42666     .00345    18     1    .7052E-04    17    24    .3017E+02    .5459E-04    36    .2904E-02    62    .1790E-03
200     .42686     .00345    18     1    .6571E-04    17    24    .2802E+02    .5117E-04    36    .2866E-02    52    .7496E-04
210     .42705     .00345    16     1    .6247E-04    17    24    .2629E+02    .4805E-04    36    .2544E-02    43    .4596E-04
220     .42722     .00346    14     1    .5839E-04    17    24    .2449E+02    .4525E-04    36    .2463E-02    57    .1685E-03
230     .42739     .00346    15     1    .5414E-04    17    24    .2293E+02    .4229E-04    36    .2265E-02    18    .5778E-04
240     .42754     .00346    18     1    .5041E-04    17    24    .2134E+02    .3988E-04    36    .2327E-02    62    .1241E-03
250     .42769     .00346    18     1    .4707E-04    17    24    .1988E+02    .3713E-04    36    .2047E-02    18    .5211E-04
260     .42782     .00346    16     1    .4479E-04    17    24    .1871E+02    .3483E-04    36    .1837E-02    18    .3307E-04
270     .42795     .00346    14     1    .4169E-04    17    24    .1746E+02    .3278E-04    36    .1753E-02    57    .7669E-04
280     .42807     .00347    21     1    .3889E-04    17    24    .1638E+02    .3062E-04    36    .1619E-02    18    .4589E-04
290     .42818     .00347    18     1    .3623E-04    17    24    .1527E+02    .2871E-04    35    .1448E-02    62    .6897E-04
300     .42828     .00347    18     1    .3386E-04    17    24    .1425E+02    .2686E-04    35    .1465E-02    18    .4630E-04
310     .42838     .00347    16     1    .3224E-04    17    24    .1343E+02    .2519E-04    35    .1297E-02    18    .2604E-04
320     .42847     .00347    14     1    .3003E-04    17    24    .1255E+02    .2370E-04    35    .1258E-02    57    .3567E-04
330     .42856     .00347    21     1    .2802E-04    17    24    .1178E+02    .2213E-04    35    .1164E-02    18    .3435E-04
340     .42864     .00347    18     1    .2612E-04    17    24    .1099E+02    .2075E-04    35    .1341E-02    62    .6419E-04
350     .42872     .00347    18     1    .2441E-04    17    24    .1026E+02    .1941E-04    36    .1355E-02    18    .2982E-04
360     .42879     .00347    16     1    .2325E-04    17    24    .9671E+01    .1820E-04    36    .9338E-03    18    .1934E-04
370     .42885     .00347    14     1    .2166E-04    17    24    .9042E+01    .1713E-04    35    .9358E-03    57    .3978E-04
380     .42892     .00347    21     1    .2022E-04    17    24    .8494E+01    .1599E-04    35    .8373E-03    18    .2516E-04
390     .42897     .00347    18     1    .1884E-04    17    24    .7924E+01    .1499E-04    36    .7513E-03    62    .4641E-04
400     .42903     .00348    18     1    .1762E-04    17    24    .7400E+01    .1403E-04    36    .7616E-03    18    .2174E-04
410     .42908     .00348    16     1    .1678E-04    17    24    .6977E+01    .1319E-04    36    .6742E-03    18    .1416E-04
420     .42913     .00348    14     1    .1563E-04    17    24    .6529E+01    .1237E-04    35    .6541E-03    57    .2877E-04
430     .42917     .00348    21     1    .1460E-04    17    24    .6130E+01    .1159E-04    36    .6262E-03    18    .1825E-04
440     .42922     .00348    18     1    .1360E-04    17    24    .5720E+01    .1083E-04    35    .5426E-03    62    .3359E-04
450     .42925     .00348    18     1    .1272E-04    17    24    .5342E+01    .1018E-04    36    .5502E-03    18    .1575E-04
460     .42929     .00348    16     1    .1212E-04    17    24    .5037E+01    .9499E-05    35    .4873E-03    18    .1621E-04
470     .42933     .00348    14     1    .1129E-04    17    24    .4711E+01    .8930E-05    36    .4726E-03    57    .2682E-04
480     .42936     .00348    21     1    .1054E-04    17    24    .4426E+01    .8347E-05    35    .4383E-03    18    .1321E-04
490     .42939     .00348    18     1    .9824E-05    17    24    .4130E+01    .7826E-05    35    .3929E-03    62    .2433E-04

              ******  SOLUTION CONVERGED  ******
```

```
                                                            ******************
                                                            *                *
                                                            *  FINAL OUTPUT  *
                                                            *                *
                                                            ******************

PRINTOUT IN  PHYSICAL  VARIABLES.

DEFINITION OF SIMILARITY PARAMETERS BY KRUPP

BOUNDARY CONDITION FOR SOLID WALL

DIFFERENCE EQUATIONS ARE FULLY CONSERVATIVE.

KUTTA CONDITION IS ENFORCED.

            MACH =    .7900000
           DELTA =    .0600000
           ALPHA =   2.0000000
               K =   3.0061213
DOUBLET STRENGTH =    .0526114
  AIRFOIL VOLUME =    .0410849

      PARAMETERS USED TO TRANSFORM VARIABLES
              TO TRANSONIC SCALING

          CPFACT =    .1901683
          CDFACT =    .0114101
          CMFACT =    .1901683
          CLFACT =    .1901683
           YFACT =   2.9495264
           VFACT =   3.4377467
```

Figure 4.5.- Continued

```
                                   FINAL OUTPUT FOR FINE MESH

FORCE COEFFICIENTS, PRESSURE COEFFICIENT, AND MACH NUMBER
(OR SIMILARITY PARAMETER) ON BODY AND DIVIDING STREAM LINE.
                                                                          B = BEFORE OR BEHIND AIRFOIL
        CL =    .429390                                                   U = UPPER AIRFOIL SURFACE
        CM =    .003479                                                   L = LOWER AIRFOIL SURFACE
       CP* =  -.603170                                                    * = CRITICAL PRESSURE

                         LOWER                          UPPER
                         Y=0-                           Y=0+
  I        X          CP          M1               CP          M1

  1   -1.075000     .008699     .745782          .008699     .745782
  2    -.950000     .016709     .744765          .016709     .744765
  3    -.825000     .013955     .743416          .013955     .743416
  4    -.700000     .018026     .741232          .018026     .741232
  5    -.575000     .025241     .737693          .025241     .737693
  6    -.450000     .036605     .732086          .036605     .732086
  7    -.350000     .053754     .723540          .053754     .723540
  8    -.250000     .080660     .709922          .080660     .709922
  9    -.175000     .119291     .689909          .119291     .689909
 10    -.125000     .171516     .661886          .171516     .661886
 11    -.075000     .246613     .619373          .246613     .619373
 12    -.052500     .331554     .567461          .331554     .567461
 13    -.035000     .419085     .508452          .419085     .508452
 14    -.022500     .511797     .437351          .511797     .437351
 15    -.015000     .597950     .358868          .597950     .358868
 16    -.007500     .678353     .265456          .678353     .265456
 17    -.002500     .740611     .139665          .740611     .139665
                     AIRFOIL LEADING EDGE
 18     .002500    1.322901    0.000000          .076092     .712276
 19     .007500     .946407    0.000000        -.456347     .949253
 20     .012500     .729237     .183192        -.807212    1.071447
 21     .017500     .579391     .377158       -1.005689    1.136645
 22     .022500     .470207     .470577       -1.024078    1.142497
 23     .027500     .388169     .530045       -1.046277    1.149522
 24     .032500     .325669     .571591       -1.074236    1.158309
 25     .037500     .269801     .605644       -1.089525    1.163277
 26     .045000     .213693     .638358       -1.077035    1.159185
 27     .055000     .162263     .666937       -1.071093    1.157324
 28     .065000     .122016     .688475       -1.067760    1.156280
 29     .075000     .091169     .704537       -1.066786    1.155976
 30     .085000     .065166     .717797       -1.065965    1.155711
 31     .097500     .039543     .730629       -1.062600    1.154686
 32     .115000     .012083     .744134       -1.054815    1.152213
 33     .140625    -.019640     .757241       -1.042097    1.148262
 34     .171875    -.037047     .767705       -1.025352    1.142901
 35     .203125    -.052108     .774788       -1.001581    1.135333
 36     .234375    -.061703     .779266        -.849011    1.084467
 37     .265625    -.067407     .781916        -.519745     .967774
 38     .296875    -.070239     .783220        -.366674     .932196
 39     .328125    -.070966     .783537        -.370809     .911045
 40     .359375    -.069920     .783081        -.371987     .912154
 41     .390625    -.067660     .782033        -.360922     .907904
 42     .421875    -.064411     .780525        -.344785     .901435
 43     .453125    -.060386     .778653        -.326131     .893886
 44     .484375    -.055747     .776489        -.306299     .885815
 45     .515625    -.050613     .774087        -.286087     .877501
 46     .546875    -.045666     .771484        -.265806     .869079
 47     .578125    -.039155     .768700        -.245599     .860605
 48     .609375    -.032899     .765743        -.225498     .852092
 49     .640625    -.026209     .762600        -.205450     .843523
 50     .671875    -.019202     .759267        -.185881     .834843
 51     .703125    -.011860     .755685        -.165116     .825993
 52     .734375    -.003719     .751796        -.144455     .816871
 53     .765625     .005140     .747507        -.123119     .807343
 54     .796875     .015074     .742675        -.100721     .797210
 55     .828125     .026472     .737088        -.076713     .786220
 56     .859375     .039336     .730731        -.051494     .774500
 57     .885000     .051919     .724459        -.030437     .763627
 58     .900000     .062681     .719051        -.009777     .754713
 59     .915000     .073340     .713655         .007091     .746175
 60     .930000     .085997     .707194         .027957     .736397
 61     .945000     .101977     .699159         .051517     .724660
 62     .960000     .121071     .688551         .080671     .709920
 63     .975000     .150702     .673194         .119856     .689612
 64     .990000     .192920     .650271         .173673     .650793
 65    1.000000     .249123     .617902         .243370     .621269
                     AIRFOIL TRAILING EDGE
 66    1.010000     .219841     .637137          .219841     .637137
 67    1.025000     .165663     .665085          .165663     .665085
 68    1.050000     .123379     .687757          .123379     .687757
 69    1.090000     .089664     .705311          .089664     .705311
 70    1.150000     .064386     .718191          .064386     .718191
 71    1.225000     .047177     .726829          .047177     .726829
 72    1.300000     .035460     .732653          .035460     .732653
 73    1.400000     .026855     .736900          .026855     .736900
 74    1.500000     .020505     .739979          .020505     .739979
 75    1.625000     .015970     .742237          .015970     .742237
 76    1.750000     .012765     .743802          .012765     .743802
 77    1.875000     .010204     .745050          .010204     .745050
```

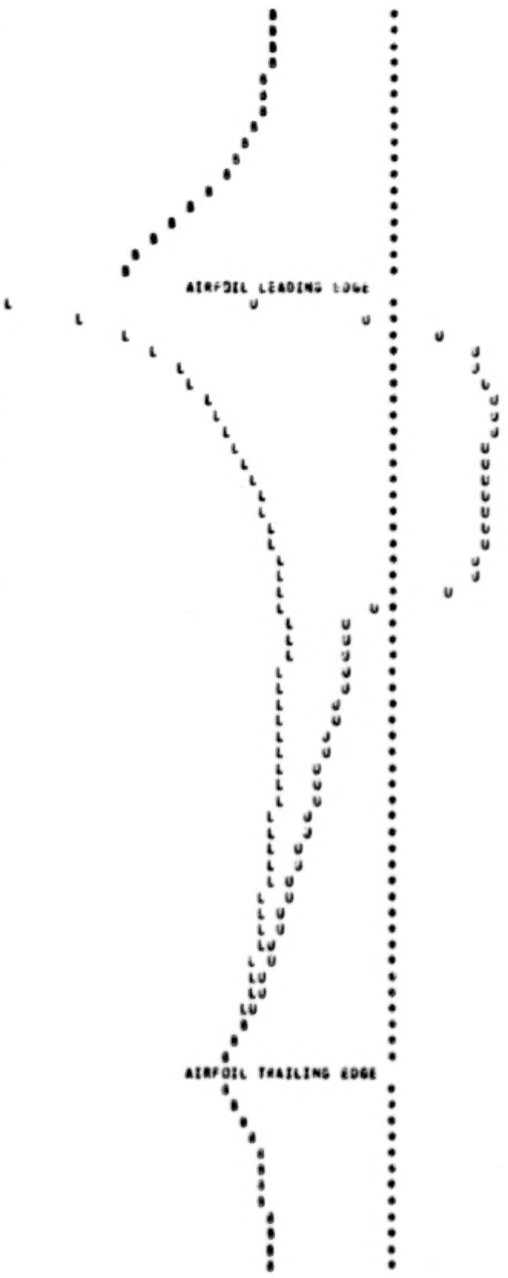


Figure 4.5.- Continued

```
     Y-GRID VALUES     Y(J) J=   1 TO  48
  -2.000000   -1.800000   -1.600000   -1.400000   -1.200000   -1.000000    -.800000    -.650000
   -.550000    -.450000    -.390000    -.340000    -.300000    -.270000    -.240000    -.210000
   -.180000    -.150000    -.125000    -.100000    -.075000    -.050000    -.030000    -.010000
    .010000     .030000     .050000     .075000     .100000     .125000     .150000     .180000
    .210000     .240000     .270000     .300000     .340000     .390000     .450000     .550000
    .650000     .800000    1.000000    1.200000    1.400000    1.600000    1.800000    2.000000
```

```
                                   FLOW CHARACTER MAP
                                   ******************

                    P = PARABOLIC
                    H = HYPERBOLIC
                    S = SHOCK
                    - = ELLIPTIC

 48      ------------------------------------------------------------------------------------------------
 47      ------------------------------------------------------------------------------------------------
 46      ------------------------------------------------------------------------------------------------
 45      ------------------------------------------------------------------------------------------------
 44      ------------------------------------------------------------------------------------------------
 43      ------------------------------------------------------------------------------------------------
 42      ------------------------------------------------------------------------------------------------
 41      ------------------------------------------------------------------------------------------------
 40      ------------------------------------------------------------------------------------------------
 39      ------------------------------------------------------------------------------------------------
 38      ------------------------------------------------------------------------------------------------
 37      ------------------------------------------------------------------------------------------------
 36      ---------------------------------------PS-------------------------------------------------------
 35      --------------------------------------PHHS------------------------------------------------------
 34      --------------------------------------PHHS------------------------------------------------------
 33      -------------------------------------PHHHS------------------------------------------------------
 32      ------------------------------------PHHHHHS-----------------------------------------------------
 31      -----------------------------------PHHHHHHS-----------------------------------------------------
 30      ----------------------------------PHHHHHHHS-----------------------------------------------------
 29      ---------------------------------PHHHHHHHHS-----------------------------------------------------
 28      -------------------------------PHHHHHHHHHHS-----------------------------------------------------
 27      ------------------------------PHHHHHHHHHHHS-----------------------------------------------------
 26      ----------------------------PHHHHHHHHHHHHHS-----------------------------------------------------
 25      --------------------------PHHHHHHHHHHHHHHHS-----------------------------------------------------
 24      ------------------------------------------------------------------------------------------------
 23      ------------------------------------------------------------------------------------------------
 22      ------------------------------------------------------------------------------------------------
 21      ------------------------------------------------------------------------------------------------
 20      ------------------------------------------------------------------------------------------------
 19      ------------------------------------------------------------------------------------------------
 18      ------------------------------------------------------------------------------------------------
 17      ------------------------------------------------------------------------------------------------
 16      ------------------------------------------------------------------------------------------------
 15      ------------------------------------------------------------------------------------------------
 14      ------------------------------------------------------------------------------------------------
 13      ------------------------------------------------------------------------------------------------
 12      ------------------------------------------------------------------------------------------------
 11      ------------------------------------------------------------------------------------------------
 10      ------------------------------------------------------------------------------------------------
  9      ------------------------------------------------------------------------------------------------
  8      ------------------------------------------------------------------------------------------------
  7      ------------------------------------------------------------------------------------------------
  6      ------------------------------------------------------------------------------------------------
  5      ------------------------------------------------------------------------------------------------
  4      ------------------------------------------------------------------------------------------------
  3      ------------------------------------------------------------------------------------------------
  2      ------------------------------------------------------------------------------------------------
  1      ------------------------------------------------------------------------------------------------
```

```
                                         MACH NUMBER MAP
                                         ***************

SYMBOL       RANGE             SYMBOL       RANGE             SYMBOL       RANGE             SYMBOL       RANGE
   0             M.LE.0.05        A   0.95.LT.M.LE.1.05        K   1.95.LT.M.LE.2.05        U   2.95.LT.M.LE.3.05
   1   0.05.LT.M.LE.0.15          B   1.05.LT.M.LE.1.15        L   2.05.LT.M.LE.2.15        V   3.05.LT.M.LE.3.15
   2   0.15.LT.M.LE.0.25          C   1.15.LT.M.LE.1.25        M   2.15.LT.M.LE.2.25        W   3.15.LT.M.LE.3.25
   3   0.25.LT.M.LE.0.35          D   1.25.LT.M.LE.1.35        N   2.25.LT.M.LE.2.35        X   3.25.LT.M.LE.3.35
   4   0.35.LT.M.LE.0.45          E   1.35.LT.M.LE.1.45        O   2.35.LT.M.LE.2.45        Y   3.35.LT.M.LE.3.45
   5   0.45.LT.M.LE.0.55          F   1.45.LT.M.LE.1.55        P   2.45.LT.M.LE.2.55        Z   3.45.LT.M.LE.3.55
   6   0.55.LT.M.LE.0.65          G   1.55.LT.M.LE.1.65        Q   2.55.LT.M.LE.2.65        *             M.GT.3.55
   7   0.65.LT.M.LE.0.75          H   1.65.LT.M.LE.1.75        R   2.65.LT.M.LE.2.75
   8   0.75.LT.M.LE.0.85          I   1.75.LT.M.LE.1.85        S   2.75.LT.M.LE.2.85
   9   0.85.LT.M.LE.0.95          J   1.85.LT.M.LE.1.95        T   2.85.LT.M.LE.2.95
```

Figure 4.5.- Continued

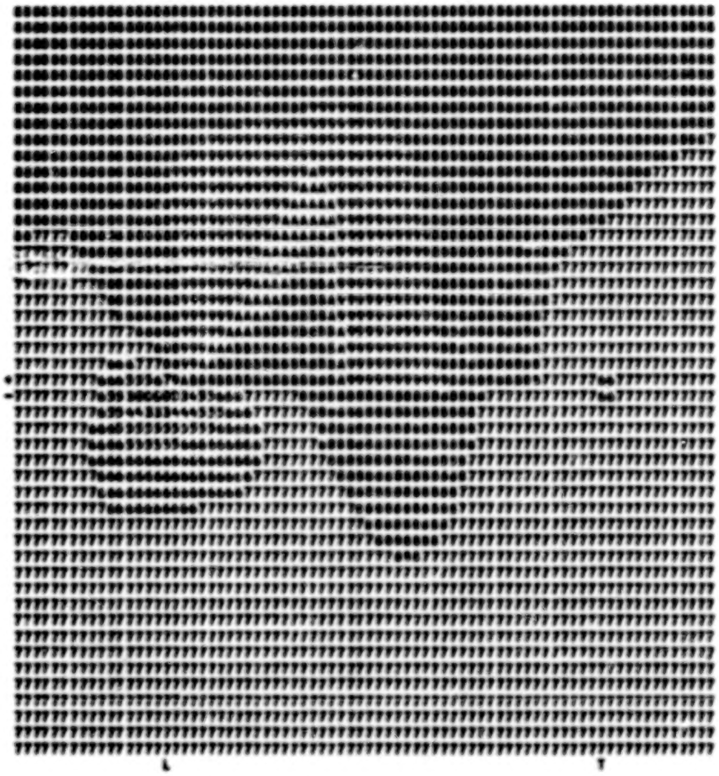


Figure 4.5.- Continued

PRINTER PLOT OF CP ON BODY AND DIVIDING STREAMLINE

```
  U = CP(UPPER)
  L = CP(LOWER)
  B = CP(UPPER) = CP(LOWER)
--- = CRITICAL CP
```

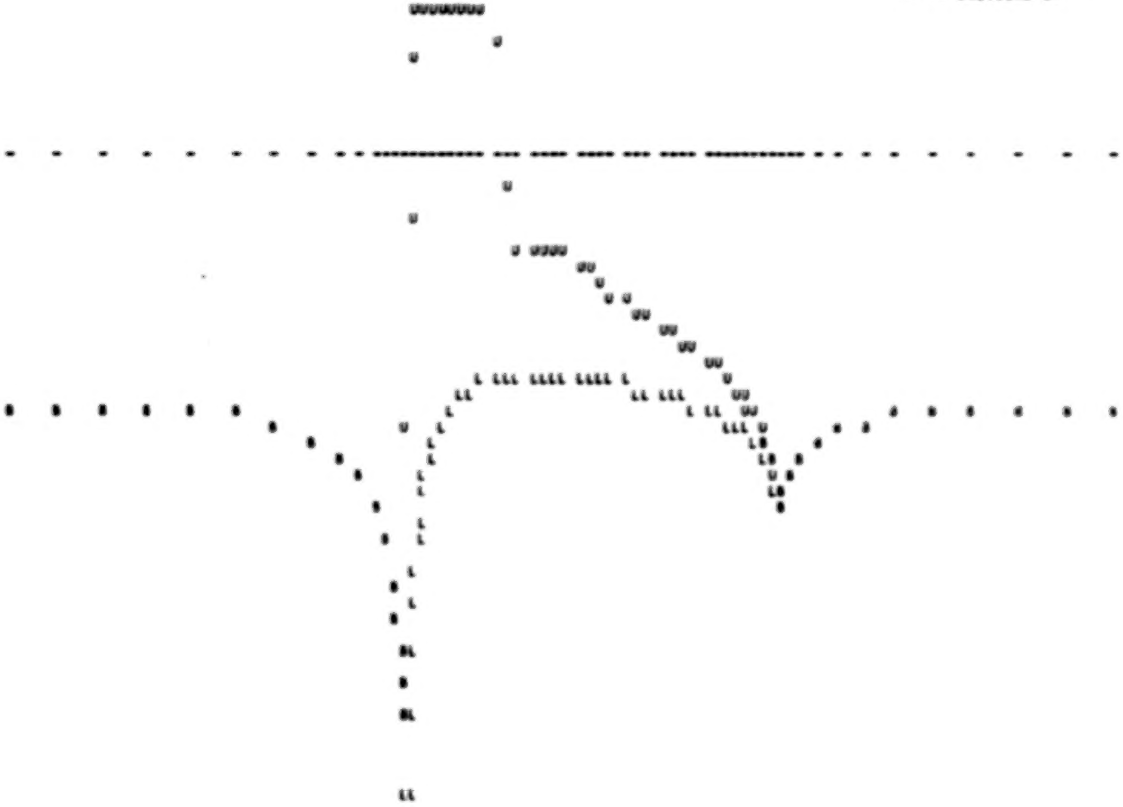


Figure 4.5.- Continued

```
SOLID WALL BOUNDARY CONDITION.

        H (TUNNEL HALF HEIGHT) = 2.000000                    SCALED H =  .678075

                             CP* = -.633176

                               LOWER                          UPPER
                               Y=-H                           Y=+H
 I        X          CP         THETA              CP          THETA

 2    -.950000    .035447    0.000000          -.048436    0.000000
 3    -.825000    .037896    0.000000          -.054969    0.000000
 4    -.700000    .040216    0.000000          -.062600    0.000000
 5    -.575000    .042319    0.000000          -.071347    0.000000
 6    -.450000    .044013    0.000000          -.080504    0.000000
 7    -.350000    .045193    0.000000          -.089420    0.000000
 8    -.250000    .045926    0.000000          -.097874    0.000000
 9    -.175000    .046272    0.000000          -.103624    0.000000
10    -.125000    .046393    0.000000          -.108106    0.000000
11    -.075000    .046424    0.000000          -.111338    0.000000
12    -.052500    .046426    0.000000          -.113315    0.000000
13    -.035000    .046423    0.000000          -.114460    0.000000
14    -.022500    .046427    0.000000          -.115272    0.000000
15    -.015000    .046502    0.000000          -.115802    0.000000
16    -.007500    .046558    0.000000          -.116193    0.000000
17    -.002500    .046618    0.000000          -.116499    0.000000
18     .002500    .046618    0.000000          -.116793    0.000000
19     .007500    .046624    0.000000          -.117086    0.000000
20     .012500    .046616    0.000000          -.117374    0.000000
21     .017500    .046605    0.000000          -.117657    0.000000
22     .022500    .046599    0.000000          -.117936    0.000000
23     .027500    .046592    0.000000          -.118211    0.000000
24     .032500    .046584    0.000000          -.118481    0.000000
25     .037500    .046511    0.000000          -.118810    0.000000
26     .045000    .046392    0.000000          -.119240    0.000000
27     .055000    .046319    0.000000          -.119739    0.000000
28     .065000    .046260    0.000000          -.120220    0.000000
29     .075000    .046213    0.000000          -.120697    0.000000
30     .085000    .046132    0.000000          -.121161    0.000000
31     .097500    .046007    0.000000          -.121754    0.000000
32     .115000    .045836    0.000000          -.122479    0.000000
33     .140625    .045612    0.000000          -.123327    0.000000
34     .171875    .045341    0.000000          -.124126    0.000000
35     .203125    .045037    0.000000          -.124714    0.000000
36     .234375    .044707    0.000000          -.125044    0.000000
37     .265625    .044354    0.000000          -.125113    0.000000
38     .296875    .043982    0.000000          -.124916    0.000000
39     .328125    .043593    0.000000          -.124455    0.000000
40     .359375    .043109    0.000000          -.123733    0.000000
41     .390625    .042776    0.000000          -.122760    0.000000
42     .421875    .042338    0.000000          -.121547    0.000000
43     .453125    .041895    0.000000          -.120104    0.000000
44     .484375    .041444    0.000000          -.118445    0.000000
45     .515625    .040984    0.000000          -.116505    0.000000
46     .546875    .040517    0.000000          -.114342    0.000000
47     .578125    .040046    0.000000          -.112330    0.000000
48     .609375    .039570    0.000000          -.109867    0.000000
49     .640625    .039091    0.000000          -.107460    0.000000
50     .671875    .038610    0.000000          -.104855    0.000000
51     .703125    .038125    0.000000          -.102141    0.000000
52     .734375    .037638    0.000000          -.099344    0.000000
53     .765625    .037147    0.000000          -.096481    0.000000
54     .796875    .036652    0.000000          -.093566    0.000000
55     .828125    .036155    0.000000          -.090612    0.000000
56     .859375    .035656    0.000000          -.087767    0.000000
57     .885000    .035321    0.000000          -.085425    0.000000
58     .900000    .035062    0.000000          -.083725    0.000000
59     .915000    .034822    0.000000          -.082286    0.000000
60     .930000    .034581    0.000000          -.080849    0.000000
61     .945000    .034329    0.000000          -.079424    0.000000
62     .960000    .034078    0.000000          -.078002    0.000000
63     .975000    .033840    0.000000          -.076575    0.000000
64     .990000    .033649    0.000000          -.075253    0.000000
65    1.000000    .033496    0.000000          -.074181    0.000000
66    1.010000    .033284    0.000000          -.073129    0.000000
67    1.025000    .032962    0.000000          -.071619    0.000000
68    1.050000    .032482    0.000000          -.069191    0.000000
69    1.090000    .031747    0.000000          -.065429    0.000000
70    1.150000    .030694    0.000000          -.060213    0.000000
71    1.225000    .029391    0.000000          -.054143    0.000000
72    1.300000    .027857    0.000000          -.047599    0.000000
73    1.400000    .026428    0.000000          -.043877    0.000000
74    1.500000    .023890    0.000000          -.033290    0.000000
75    1.625000    .021453    0.000000          -.029961    0.000000
76    1.750000    .018008    0.000000          -.018987    0.000000
```

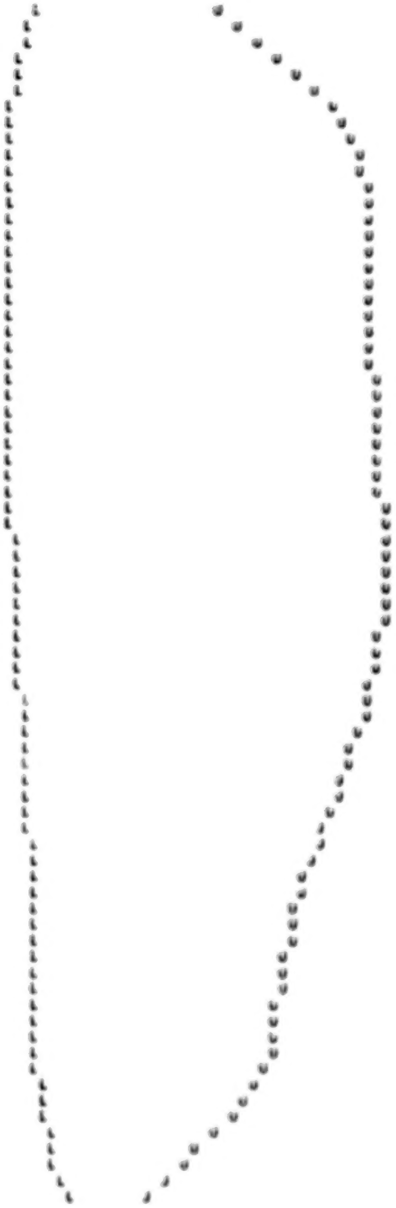


Figure 4.5.- Continued

```
                                          SONIC LINE OUTPUT
                                          *****************
SONIC LINE COORDINATES

     Y          X-SONIC

   .30000     .17004     .17632
   .27000     .13900     .21245
   .24000     .12090     .22340
   .21000     .10953     .23204
   .18000     .09171     .23862
   .15000     .07848     .24356
   .12500     .06694     .24689
   .10000     .05946     .24968
   .07500     .04410     .25205
   .05000     .03260     .25400
   .03000     .02244     .25549
   .01000     .01172     .25674
  BODY LOCATION
```

```
                                          SONIC LINE OUTPUT
                                          *****************
```

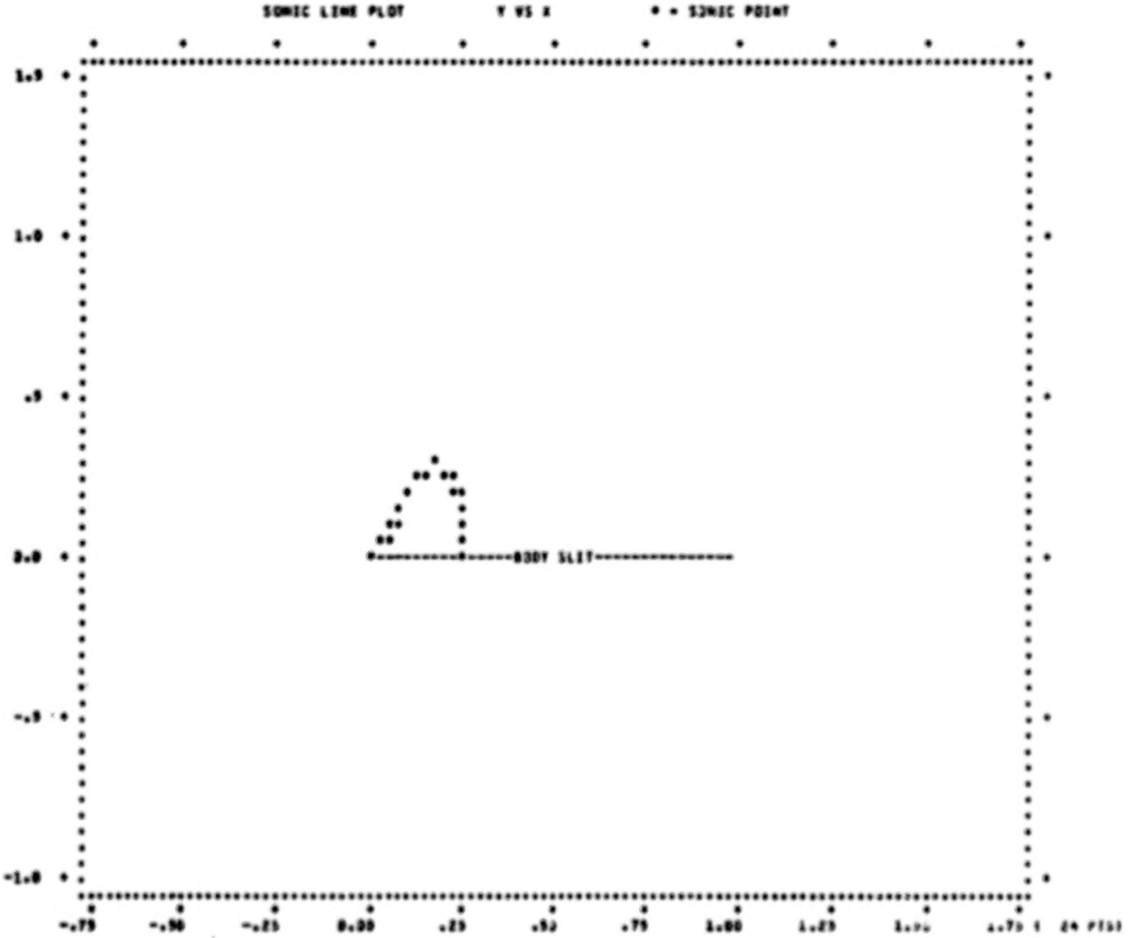


Figure 4.5.- Continued

```
                                   SHOCK WAVE DRAG AND TOTAL PRESSURE PROFILE OUTPUT
                                   *************************************************

INVISCID WAKE PROFILES FOR INDIVIDUAL SHOCK WAVES WITHIN MOMENTUM CONTOUR

SHOCK  1
WAVE DRAG FOR THIS SHOCK=      .001050
     Y           CD(Y)          PO/POINF

  0.00000000    .00790089     .99465570
   .01250000    .00769835     .99490512
   .02500000    .00730523     .99524120
   .05000000    .00677634     .99558573
   .07500000    .00597963     .99610486
   .10000000    .00509296     .99660233
   .12500000    .00417618     .99727094
   .15000000    .00320669     .99785097
   .18000000    .00232634     .99840457
   .21000000    .00102329     .99913346
   .24000000    .00066724     .99956534
   .27000000    .00040805     .99973419
   .30000000    .00020095     .99999417
```

```
                                                              DRAG COEFFICIENT OUTPUT
                                                              ***********************

CALCULATION OF DRAG COEFFICIENT BY MOMENTUM INTEGRAL METHOD

BOUNDARIES OF CONTOUR USED                  CONTRIBUTION TO CD
UPSTREAM     X =     -.175000                CDUP    =     -.000035
DOWNSTREAM   X =     1.225000                CDDOWN  =      .003219
TOP          Y =     1.000000                CDTOP   =      .003140
BOTTOM       Y =    -1.036666                CDBOT   =      .003829
               TOTAL CONTRIBUTIONS AROUND CONTOUR    =      .003333

THERE ARE  1 SHOCKS INSIDE CONTOUR. TOTAL CDWAVE =      .001090

NOTE - ALL SHOCKS CONTAINED WITHIN CONTOUR
CDWAVE EQUALS TOTAL WAVE DRAG

DRAG CALCULATED FROM MOMENTUM INTEGRAL    CD     =      .001391
```

```
TIME TO RUN CASE WAS  22.04 SECONDS.
```

(b) Output.

Figure 4.1.- Concluded

```
SAMPLE CASE 3 - SUBSONIC-NACA0006 AIRFOIL-SLOTTED TUNNEL WALL SIMULATION
$INP
 BCTYPE=4, F=0.066, EPS=0.9, H=2.0, ALPHA=2.0, DELTA=0.06, EMACH=0.75,
 BCFOIL=1,
$END
```

(a) Input.

```
SAMPLE CASE 3 - SUBSONIC - NACA0006 AIRFOIL - SLOTTED TUNNEL WALL SIMULATION


                                    INPUT PARAMETERS
                                    ****************

EMACH =    .75000        POR =  0.00000        IMIN =    1      BCTYPE =  4         AMESH = F

DELTA =    .06000      CLSET =  0.00000       IMAXI =   77      BCFOIL =  1          PHYS = T

ALPHA =   2.00000        EPS =   .90000        JMIN =    1      PSTART =  1         PSAVE = F

   AK =   0.00000       RIGF =  0.00000       JMAXI =   48      PRTFLO =  1         KUTTA = T

  GAM =   1.40000      WCIRC =  1.00000       MAXIT =  500      IPRTER = 10           FCR = T

    F =    .06600     CVERGE =   .00001          NU =  100      SIMDEF =  3

    H =   2.00000     DVERGE =     10.0          NL =   75        ICUT =  2

   WE = 1.80,1.90,1.95

  XIN
-1.075000     -.950000     -.825000     -.700000     -.575000     -.450000     -.350000     -.250000
 -.175000     -.125000     -.075000     -.052500     -.035000     -.022500     -.015000     -.007500
 -.002500      .002500      .007500      .012500      .017500      .022500      .027500      .032500
  .037500      .047500      .055000      .065250      .075000      .089000      .097500      .115000
  .140625      .171875      .203125      .234375      .265625      .296875      .328125      .359375
  .390625      .421875      .453125      .484375      .515625      .546875      .578125      .609375
  .640625      .671875      .703125      .734375      .765625      .796875      .828125      .859375
  .885000      .900000      .915000      .930000      .945000      .960000      .975000      .990000
 1.000000     1.010000     1.025000     1.050000     1.090000     1.150000     1.225000     1.350000
 1.400000     1.500000     1.625000     1.750000     1.875000

  ZIN
-2.000000    -1.800000    -1.600000    -1.400000    -1.200000    -1.000000     -.800000     -.650000
 -.550000     -.450000     -.390000     -.340000     -.300000     -.270000     -.240000     -.210000
 -.180000     -.150000     -.125000     -.100000     -.075000     -.050000     -.030000     -.010000
  .010000      .030000      .050000      .075000      .100000      .125000      .150000      .180000
  .210000      .240000      .270000      .300000      .340000      .390000      .450000      .550000
  .650000      .800000     1.000000     1.200000     1.400000     1.600000     1.800000     2.000000
```

Figure 4-6.- Sample test case 3 - subsonic, slotted tunnel wall simulation (abbreviated output): NACA 0006 airfoil, $\delta = 0.06$, $\alpha = 2°$, $M_\infty = 0.75$, $H = 2.$, $F = 0.066$.

```
                                  INTERMEDIATE OUTPUT FOR COARSE MESH

        WE =   1.0000     EPS =     .4000     MAXIT FOR THIS MESH =  120

ITERATION    CL         CM       IERR  JERR    ERROR     IRL   JRL      BIGRL        ERCIRC     ICPU    CPURAU     ICPL     CPLRAL

   10      .15882     .00496     14     6    .3662E-01    6     1    .1173E+03    .2474E-02    14   [illegible]    17   .9356E+00
   20      .25257     .00369     13     6    .1689E-01    6     6    .2311E+03    .1392E-01     9   .1129E+03      8   .4997E-01
   30      .27626     .00252      5     3    .1299E-01    6     6    .2312E+03    .9457E-02     8   .6227E-01      6   .2493E-01
   40      .30252     .00261      4     3    .7722E-02    6     6    .1355E+03    .5715E-02     7   .8156E-01      7   .1228E-01
   50      .31826     .00275      4     3    .4441E-02    6     6    .7797E+02    .3442E-02     7   .6496E-01     16   .6528E-02
   60      .32763     .00287      4     3    .2748E-02    6     6    .4702E+02    .2205E-02     7   [illegible]   16   .2990E-02
   70      .33361     .00295      4     3    .1661E-02    6     6    .2992E+02    .1263E-02     6   .2251E-01     16   .2214E-02
   80      .33712     .00301      4     3    .1016E-02    6     6    .1768E+02    .7692E-03     6   .1130E-01     16   [illegible]
   90      .33927     .00306      4     3    .6179E-03    6     6    .1072E+02    .4646E-03     5   [illegible]   16   .8462E-03
  100      .34057     .00309      4     3    .3735E-03    6     6    .6490E+01    .2848E-03     6   [illegible]   16   .9496E-03
  110      .34136     .00310      4     3    .2292E-03    6     6    .3985E+01    .1741E-03     6   .3377E-02     16   .2877E-03
  120      .34183     .00311      4     3    .1385E-03    6     6    .2429E+01    .1107E-03     6   [illegible]   16   .1313E-03

           ******  ITERATION LIMIT REACHED  ******



                                  INTERMEDIATE OUTPUT FOR MEDIUM MESH

        WE =   1.9000     EPS =     .5000     MAXIT FOR THIS MESH =  250

ITERATION    CL         CM       IERR  JERR    ERROR     IRL   JRL      BIGRL        ERCIRC     ICPU    CPURAU     ICPL     CPLRAL

   10      .34350     .00682     12    17    .1440E-02   10    12    .1134E+03    .1769E-03    33   .1235E+01     19   .9598E+00
   20      .34382     .00646     20    13    .6586E-03   13    12    .4997E+02    .1372E-03    13   [illegible]   12   .7771E-01
   30      .34436     .00655     12     6    .4069E-03   10    12    .2603E+02    .1499E-03    14   [illegible]   30   .2813E-02
   40      .34487     .00661     12    13    .3335E-03   10    12    .2684E+02    .1326E-03    14   [illegible]   20   .1127E-02
   50      .34534     .00672     12    13    .2994E-03   10    12    .1899E+02    .1101E-03    13   [illegible]   13   .2229E-02
   60      .34576     .00683     12    13    .2442E-03   13    12    .1174E+02    .9147E-04    13   [illegible]   12   .1367E-02
   70      .34604     .00700     12    13    .1621E-03   10    12    .6772E+01    .8511E-04    13   .5204E-02     12   .7622E-03
   80      .34632     .00703      9     6    .7727E-04   10    12    .5744E+01    .6756E-04    14   [illegible]   12   .3465E-03
   90      .34654     .00703     29    12    .6318E-04   10    12    .4240E+01    .3038E-04    17   .1170E-02     10   .1572E-03
  100      .34670     .00703     25    12    .4736E-04   13    12    .3649E+01    .3917E-04    17   [illegible]   13   [illegible]
  110      .34683     .00703      8     5    .3859E-04   10    12    .3127E+01    .5662E-04    17   [illegible]   15   .4017E-04
  120      .34693     .00703      8     3    .3134E-04   13    12    .2379E+01    .2484E-04    17   [illegible]   10   .4443E-04
  130      .34700     .00703      8     6    .2481E-04   10    12    .2096E+01    .1487E-04    17   [illegible]   12   .3464E-04
  140      .34707     .00703      5     6    .1995E-04   10    12    .1614E+01    .1469E-04    17   [illegible]   12   .3144E-04
  150      .34711     .00703      8     6    .1537E-04   10    12    .1760E+01    .1178E-04    17   .2179E-02     12   .2196E-04
  160      .34715     .00703      8     6    .1210E-04   10    12    .9984E+00    .9281E-05    17   [illegible]   12   .1729E-04
  170      .34718     .00703      8     6    .9471E-05   10    12    .7850E+00    .7283E-05    17   [illegible]   12   .1465E-04

           ******  SOLUTION CONVERGED  ******



                                  INTERMEDIATE OUTPUT FOR FINE MESH

        WE =   1.9500     EPS =     .9700     MAXIT FOR THIS MESH =  400

ITERATION    CL         CM       IERR  JERR    ERROR     IRL   JRL      BIGRL        ERCIRC     ICPU    CPURAU     ICPL     CPLRAL

   10      .34704     .00843     22    21    .1502E-02   22    21    .3007E+03    .1223E-03    32   [illegible]   19   .1377E+00
   20      .34818     .00863     18    18    .6631E-03   17    24    .2742E+03    .8694E-04    33   [illegible]   24   .1442E-01
   30      .34846     .00871     16    13    .5411E-03   17    24    .1839E+03    .6781E-04    22   .2765E-01     20   .7364E-02
   40      .34876     .00876     18    10    .3373E-03   17    24    .1338E+03    .6625E-04    23   [illegible]   22   .9629E-02
   50      .34909     .00881     16     6    .2599E-03   17    24    .9948E+02    .6524E-04    24   [illegible]   21   .2729E-02
   60      .34939     .00884     22     6    .1932E-03   17    24    .7897E+02    .7474E-04    24   [illegible]   24   .1306E-02
   70      .34966     .00885     15     5    .1620E-03   17    24    .6697E+02    .6601E-04    24   [illegible]   24   .6828E-03
   80      .34991     .00887     13     6    .1336E-03   17    24    .5794E+02    .6875E-04    24   [illegible]   24   .5290E-03
   90      .35016     .00891     16    23    .1153E-03   17    24    .5222E+02    .6331E-04    24   [illegible]   24   .4447E-04
  100      .35038     .00892     16    25    .9682E-04   17    24    .4371E+02    .5444E-04    24   [illegible]   24   .7627E-03
  110      .35057     .00894     16    25    .8178E-04   17    24    .3669E+02    .4621E-04    24   [illegible]   24   .5616E-03
  120      .35073     .00895     16    25    .6700E-04   17    24    .3018E+02    .4469E-04   [illegible] [illegible] 24   .4420E-03
  130      .35090     .00895     14    23    .5552E-04   17    24    .2469E+02    .3934E-04    24   [illegible]   24   .3794E-03
  140      .35104     .00896     14    23    .4622E-04   17    24    .2086E+02    .3470E-04    24   [illegible]   24   .2390E-03
  150      .35116     .00896     14    14    .3993E-04   17    24    .1774E+02    .3263E-04    24   [illegible]   24   .1682E-03
```

Figure 4.6.- Continued

```
 160     -.35127    -.00097   18    9   -.33161E-04   17   24   -.13991E+02   -.27391E-04   34   -.14820E-02   24   -.14491E-03
 170     -.35137    -.00097   14    8   -.31566E-04   17   24   -.13600E+02   -.24221E-04   34   -.12820E-02   24   -.82931E-04
 180     -.35145    -.00097   13    9   -.28318E-04   17   24   -.12218E+02   -.21411E-04   34   -.13090E-02   24   -.94611E-04
 190     -.35153    -.00097   14   13   -.25011E-04   17   24   -.10980E+02   -.18981E-04   34   -.94295E-03   24   -.40031E-04
 200     -.35159    -.00096   14   11   -.22281E-04   17   24   -.98431E+01   -.16661E-04   34   -.87481E-03   24   -.42931E-04
 210     -.35163    -.00096   14   12   -.19941E-04   17   24   -.87964E+01   -.15036E-04   34   -.78598E-03   24   -.42301E-04
 220     -.35171    -.00096   14   13   -.17571E-04   17   24   -.77084E+01   -.13861E-04   34   -.70221E-03   24   -.41781E-04
 230     -.35175    -.00096   13   12   -.15556E-04   17   24   -.68958E+01   -.11866E-04   34   -.64201E-03   24   -.39931E-04
 240     -.35180    -.00095   14   11   -.13867E-04   17   24   -.61228E+01   -.10931E-04   34   -.56121E-03   24   -.36031E-04
 250     -.35183    -.00095   14   11   -.12581E-04   17   24   -.54360E+01   -.93481E-05   34   -.49916E-03   24   -.29111E-04
 260     -.35187    -.00095   14   11   -.10946E-04   17   24   -.48438E+01   -.82940E-05   34   -.44756E-03   24   -.21711E-04
 270     -.35190    -.00094   14   12   -.96858E-05   17   24   -.42938E+01   .73621E-05    34   -.39111E-03   24   -.22341E-04

            ******  SOLUTION CONVERGED  ******



                                                   ******************
                                                   *                *
                                                   *  FINAL OUTPUT  *
                                                   *                *
                                                   ******************

PRINTOUT IN  PHYSICAL  VARIABLES.

DEFINITION OF SIMILARITY PARAMETERS BY KRUPP

BOUNDARY CONDITION FOR SLOTTED WALL

DIFFERENCE EQUATIONS ARE FULLY CONSERVATIVE.

KUTTA CONDITION IS ENFORCED.

             MACH =     .7300000
            DELTA =     .0600000
            ALPHA =    2.0000000
                K =    2.8061713
  DOUBLET STRENGTH =     .0462323
    AIRFOIL VOLUME =     .0418099

        PARAMETERS USED TO TRANSFORM VARIABLES
                TO TRANSONIC SCALING

           CPFACT =     .1901683
           CDFACT =     .0114101
           CMFACT =     .1901683
           CLFACT =     .1901683
            YFACT =    2.9495264
            VFACT =    3.4377467
```

Figure 4.6.- Continued

```
                                        FINAL OUTPUT FOR FINE MESH

FORCE COEFFICIENTS, PRESSURE COEFFICIENT, AND MACH NUMBER
(OR SIMILARITY PARAMETER) ON BODY AND DIVIDING STREAM LINE.
                                                                      B = BEFORE OR BEHIND AIRFOIL
      CL  =   [illegible]                                             U = UPPER AIRFOIL SURFACE
      CM  =   [illegible]                                             L = LOWER AIRFOIL SURFACE
      CP* =  [illegible]                                              * = CRITICAL PRESSURE

                           LOWER                       UPPER
                           Y=0-                        Y=0+
  I        X          CP          M1            CP          M1

  1   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
  2   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
  3   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
  4   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
  5   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
  6   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
  7   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
  8   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
  9   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 10   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 11   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 12   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 13   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 14   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 15   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 16   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 17   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
                       AIRFOIL LEADING EDGE
 18   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 19   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 20   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 21   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 22   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 23   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 24   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 25   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 26   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 27   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 28   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 29   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 30   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 31   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 32   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 33   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 34   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 35   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 36   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 37   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 38   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 39   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 40   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 41   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 42   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 43   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 44   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 45   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 46   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 47   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 48   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 49   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 50   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 51   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 52   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 53   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 54   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 55   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 56   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 57   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 58   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 59   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 60   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 61   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 62   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 63   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 64   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 65   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
                       AIRFOIL TRAILING EDGE
 66   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 67   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 68   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 69   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 70   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 71   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 72   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 73   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 74   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 75   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 76   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
 77   [illegible]  [illegible]  [illegible]  [illegible]  [illegible]
```

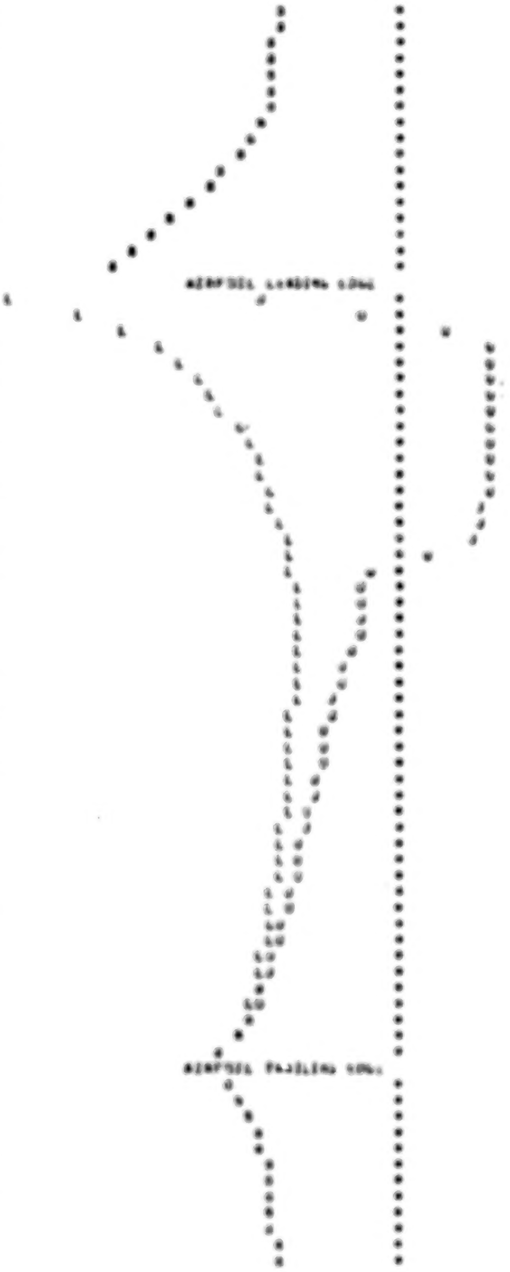


Figure 4.6.- Continued

```
    Y-GRID VALUES      Y(J) J=   1 TO  48
-2.000000   -1.800000   -1.600000   -1.400000   -1.250000   -1.000000    -.800000    -.650000
 -.550000    -.450000    -.390000    -.340000    -.300000    -.270000    -.240000    -.213333
 -.183333    -.150000    -.125000    -.100000    -.075000    -.050000    -.030000    -.010000
  .010000     .030000     .050000     .075000     .100000     .125000     .150000     .183333
  .210000     .240000     .270000     .300000     .340000     .390000     .450000     .533333
  .650000     .800000    1.000000    1.250000    1.400000    1.600000    1.800000    2.000000



                                                  SONIC LINE OUTPUT
                                                  *****************
SONIC LINE COORDINATES

     Y           X-SONIC

   .18333      .11878      .13374
   .15000      .08784      .15473
   .12500      .07528      .16011
   .10000      .05973      .16809
   .07500      .04625      .17303
   .05000      .03394      .17783
   .03000      .02284      .18050
   .01000      .01264      .18584
  BODY LOCATION



                                   SHOCK WAVE DRAG AND TOTAL PRESSURE PROFILE OUTPUT
                                   *************************************************
INVISCID WAKE PROFILES FOR INDIVIDUAL SHOCK WAVES WITHIN MOMENTUM CONTOUR

SHOCK   1
WAVE DRAG FOR THIS SHOCK=      .000342
      Y              CD(Y)          P0/P0INF

  0.00000000     .00187963      .99767731
   .01000000     .00360048      .99760245
   .03000000     .00136232      .99784679
   .05000000     .00294080      .99814496
   .07500000     .00226718      .99876219
   .10000000     .00162955      .99913067
   .12500000     .00102954      .99953324
   .15000000     .00058813      .99961687
   .18000000     .00012994      .99990194



                                                      DRAG COEFFICIENT OUTPUT
                                                      ***********************
CALCULATION OF DRAG COEFFICIENT BY MOMENTUM INTEGRAL METHOD

BOUNDARIES OF CONTOUR USED                 CONTRIBUTION TO CD
UPSTREAM     X =     -.175000              CDUP    =      -.001610
DOWNSTREAM   X =     1.225000              CDDOWN  =       .251962
TOP          Y =     1.800000              CDTOP   =       .040100
BOTTOM       Y =    -1.800000              CDBOT   =       .090330
                 TOTAL CONTRIBUTIONS AROUND CONTOUR =      .033329

THERE ARE  1 SHOCKS INSIDE CONTOUR, TOTAL CDWAVE =      .000342

NOTE - ALL SHOCKS CONTAINED WITHIN CONTOUR
CDWAVE EQUALS TOTAL WAVE DRAG

DRAG CALCULATED FROM MOMENTUM INTEGRAL     CD     =      .000397


TIME TO RUN CASE WAS  16.41 SECONDS.
```

(b) Output.

Figure 4.6.- Concluded

```
SAMPLE CASE 4 - SUPERSONIC-NACA0006 AIRFOIL-FREE AIR SIMULATION
$INP
 EMACH=1.1, BCTYPE=1, AMESH=.TRUE., BCFOIL=1, ALPHA=2.0,
 DELTA=0.06,
$END
```

(a) Input.

```
SAMPLE CASE 4 - SUPERSONIC - NACA0006 AIRFOIL - FREE AIR SIMULATION


                              INPUT PARAMETERS
                              ****************

EMACH =  1.10000      POR =  0.00000      IMIN =   1    BCTYPE =  1        AMESH = T
DELTA =   .06000    CLSET =  0.00000      IMAX =  81    BCFOIL =  1         PHYS = T
ALPHA =  2.00000      EPS =   .20000      JMIN =   1    PSTART =  1        PSAVE = F
   AK =  0.00000     RIGF =  0.00000      JMAX =  64    PRTFLO =  1        KUTTA = T
  GAM =  1.40000    WCIRC =  1.00000     MAXIT = 500    IPRTER = 10          FCR = T
    F =  0.00000   CVERGE =   .00001        NU = 100    SIMDEF =  1
    H =  0.00000   DVERGE =     10.0        NL =  75      ICUT =  2
   WE = 1.80,1.90,1.95

 XIN
[illegible]

 YIN
[illegible]
```

Figure 4-7.- Sample test case 4 - supersonic, free air simulation (abbreviated output): NACA 0006 airfoil, δ = 0.06, α = 2°
M_∞ = 1.1

```
                                        INTERMEDIATE OUTPUT FOR COURSE MESH

         WE =   1.0000      EPS =     .2000      MAXIT FOR THIS MESH =  125

ITERATION     CL          CM        IERR  JERR      ERROR      IRL   JRL       BIGRL         ERCIRC      ICPU      CPSARU      ICPL      CPERRL

    10      .20042     -.03457      21     8     .2493E+00     5     8     .1462E+03     .1908E+03     15    [illegible]      5    .8323E+01
    20      .24402     -.04272      21     8     .5993E+00    19     8     .2942E+03     .3796E+03     17    .1206E+01     17    .2732E+01
    30      .24571     -.04315      21     8     .6895E+00    19     8     .3269E+03     .3788E+00      7    .1068E-01     17    .8869E-01
    40      .24593     -.04329      21     8     .7262E+00    19     8     .3450E+03     .3969E+00      7    .9443E-02     17    .1759E-01
    50      .24609     -.04337      21     8     .7445E+00    19     8     .3546E+03     .4202E+00      7    .5387E-02     17    .9894E-02
    60      .24628     -.04340      21     8     .7497E+00    19     8     .3590E+03     .4203E+00      7    [illegible]   17    .3502E-02
    70      .24651     -.04343      21     8     .7659E+00    19     8     .3629E+03     .4197E+00      7    [illegible]   17    .3323E-02
    80      .24675     -.04344      21     8     .7694E+00    19     8     .3646E+03     .4217E+00      7    .1377E-02     17    .2061E-02
    90      .24699     -.04345      21     8     .7714E+00    19     8     .3656E+03     .4229E+00      7    .6639E-03     17    .1394E-02
   100      .24722     -.04345      21     8     .7726E+00    19     8     .3662E+03     .4236E+00      7    .4276E-03     17    .9847E-03
   110      .24746     -.04346      21     8     .7733E+00    19     8     .3666E+03     .4241E+00      7    .2017E-03     17    .7625E-03
   120      .24769     -.04347      21     8     .7737E+00    19     8     .3669E+03     .4243E+00      7    .2142E-03     17    .3758E-03

                 ******  ITERATION LIMIT REACHED  ******




                                        INTERMEDIATE OUTPUT FOR MEDIUM MESH

        WE =   1.9000      EPS =     .2000      MAXIT FOR THIS MESH =  250

ITERATION     CL          CM        IERR  JERR      ERROR      IRL   JRL       BIGRL         ERCIRC      ICPU      CPSARU      ICPL      CPERRL

    10      .25335     -.04697       7     3     .7331E-02    13     4     .0977E+02     .1117E-02      9    .9319E+00      9    .1111E+01
    20      .23430     -.04652       3     7     .3565E-02     9    16     .8966E+02     .1380E-02     12    .1794E-01     11    .7926E-01
    30      .20844     -.04675       3    31     .2810E-02     9    16     .6194E+02     .5154E-03     12    .1128E-01      9    .2504E-02
    40      .23930     -.04707       9     5     .2210E-02     9    16     .6366E+02     .2110E-03  [illegible] .7567E-03   11    .2620E-02
    50      .20999     -.04086      11    17     .1931E-02     9    16     .7515E+02     .5549E-04  [illegible] [illegible] [illegible] [illegible]
    60      .22964     -.04084      11    17     .1660E-02     9    16     .6452E+02     .3122E-03     12    .6797E-02     11    .2229E-02
    70      .20961     -.04083      11    17     .1424E-02     9    16     .9472E+02     .1176E-04     12    .9714E-02     11    .2114E-02
    80      .20957     -.04082      11    17     .1274E-02     9    16     .4812E+02     .1733E-04  [illegible] .3368E-02   11    .4974E-02
    90      .2095[illegible] -.04081 11  17     .1013E-02     9    16     .3871E+02     .1911E-04     12    .4262E-02     11    .1734E-02
   100      .20945     -.04080      11    17     .6401E-03     9    16     .3242E+02     .2291E-04     12    .3760E-02     11    .1400E-02
   110      .20938     -.04078      11    17     .7136E-03     9    16     .2717E+02     .5616E-04     12    .7998E-02     11    .2294E-02
   120      .22929     -.04075      11    17     .5961E-03     9    16     .2264E+02     .3790E-04     12    .2044E-02     11    .1104E-02
   130      .20914     -.04070      20    31     .3955E-03     9    16     .1997E+02     .9012E-04     11    .2100E-02     11    .9963E-03
   140      .20896     -.04081      20    31     .6627E-03     9    16     .1598E+02     .7571E-03  [illegible] .6091E-02   11    .7100E-03
   150      .23053     -.04046  [illegible] 31  .4417E-03     9    16     .1329E+02     .1192E-03     12    .1290E-02     11    .3244E-03
   160      .23032     -.04043      11    31     .3160E-03     9    16     .1122E+02     .5937E-04     11    .7936E-03     11    .9417E-03
   170      .20820     -.04043      11    17     .2534E-03     9    16     .9875E+01     .3379E-04     12    [illegible]   11    .4700E-03
   180      .20815     -.04044      11    17     .2140E-03     9    16     .8136E+01     .1120E-04     11    .2391E-03     11    .4237E-03
   190      .20813     -.04045      11    17     .7960E-03     9    16     .6772E+01     [illegible]   11    .5430E-03     11    .3540E-03
   200      .20813     -.04045      11    17     .4746E-03     9    16     .5597E+01     [illegible]   12    .8906E-03     11    .2621E-03
   210      .20813     -.04045      11    17     .1222E-03     9    16     .4639E+01     .1731E-05     12    .5592E-03     11    .2230E-03
   220      .20812     -.04045      11    17     .1016E-03     9    16     .3957E+01     .1591E-05  [illegible] .4056E-03   11    .1093E-03
   230      .20812     -.04045      11    17     .8447E-04     9    16     .3206E+01     .1191E-05     12    .3989E-03  [illegible] .1549E-03
   240      .20812     -.04045      11    17     .7019E-04     9    16     .2665E+01     .9130E-06     12    .7971E-03     11    .2027E-03
   250      .20811     -.04045      11    17     .5832E-04     9    16     .2213E+01     .7523E-06     11    .2497E-04     11    .1074E-03

                 ******  ITERATION LIMIT REACHED  ******
```

Figure 4.7.- Continued

Microfilmed [illegible]
Best Available Copy

```
                                   INTERMEDIATE OUTPUT FOR FINE MESH

       WE =   1.9500     EPS =    .2900     MAXIT FOR THIS MESH =  500

ITERATION    CL        CM      IERR  JERR    ERROR      IRL   JRL     BIGRL       ERCIRC    ICPU    CPERRU     ICPL    CPERRL

    10     .20800   -.04152      6    33   .3659E-02    17    32   .4072E+03   .4022E-03    17   .1117E+01    17   .2256E+01
    20     .20842   -.04162     23     2   .2342E-02    17    32   .3603E+03   .5969E-04    22   .1493E-01    20   .6974E-02
    30     .20867   -.04169     16     2   .2189E-02    17    32   .3412E+03   .1103E-03    22   .3726E-02    21   .2390E-02
    40     .20900   -.04175     15    63   .1996E-02    17    32   .3311E+03   .1033E-03    22   .5761E-03    21   .1226E-02
    50     .20925   -.04180      7    63   .1974E-02    17    32   .3232E+03   .9063E-04    17   .4130E-03    21   .9942E-03
    60     .20964   -.04192      7    63   .2111E-02    17    32   .3177E+03   .1521E-03    63   .6498E-03    61   .7533E-03
    70     .21010   -.04203      7    63   .1849E-02    17    32   .3139E+03   .1642E-03    22   .7169E-03    55   .7985E-03
    80     .21050   -.04210      7    63   .1667E-02    17    32   .3094E+03   .1293E-03    65   .4117E-03    29   .7730E-03
    90     .21081   -.04211      7     2   .1570E-02    17    32   .3093E+03   .9688E-04    22   .5196E-03    23   .1169E-02
   100     .21102   -.04209      7     2   .1534E-02    17    32   .2898E+03   .6191E-04    22   .8146E-03    22   .1302E-02
   110     .21114   -.04206      7     2   .1408E-02    17    32   .2807E+03   .3105E-04    22   .4903E-03    23   .1237E-02
   120     .21118   -.04204     18    35   .1403E-02    17    32   .2729E+03   .8523E-05    22   .4945E-03    21   .9446E-03
   130     .21118   -.04201     18    34   .1370E-02    17    32   .2670E+03   .3906E-05    22   .5349E-03    21   .6862E-03
   140     .21116   -.04199     18    35   .1341E-02    17    32   .2617E+03   .1222E-04    22   .3408E-03    65   .3126E-03
   150     .21112   -.04197      6    59   .1318E-02    17    32   .2562E+03   .1371E-04    22   .5472E-03    65   .3111E-03
   160     .21108   -.04196      6     5   .1290E-02    17    32   .2530E+03   .1213E-04    22   .8319E-03    22   .3493E-03
   170     .21106   -.04195     17    33   .1242E-02    17    32   .2428E+03   .7663E-05    22   .4961E-03    22   .6203E-03
   180     .21105   -.04194     17    33   .1204E-02    17    32   .2353E+03   .7947E-06    22   [illegible]  27   .5567E-03
   190     .21105   -.04194      7     3   .1174E-02    17    32   .2274E+03   .2441E-05    22   .1372E-02    22   .7036E-03
   200     .21105   -.04194     17    33   .1122E-02    17    32   .2191E+03   .4260E-05    22   .1119E-02    22   .7487E-03
   210     .21107   -.04194     17    33   .1076E-02    17    32   .2103E+03   .1089E-04    22   [illegible]  22   .8344E-03
   220     .21112   -.04195     17    33   .1029E-02    17    32   .2011E+03   .1946E-04    22   .1140E-02    22   .8817E-03
   230     .21119   -.04195     17    33   .9823E-03    17    32   .1919E+03   .2639E-04    22   [illegible]  22   .9117E-03
   240     .21128   -.04196     17    33   .9361E-03    17    32   .1829E+03   .3127E-04    22   [illegible]  22   .9427E-03
   250     .21139   -.04198     18    33   .8913E-03    17    32   .1741E+03   .4102E-04    22   .8841E-03    22   .9138E-03
   260     .21152   -.04199     18    35   .8488E-03    17    32   .1657E+03   .4487E-04    22   .7451E-03    22   .6937E-03
   270     .21167   -.04200     19    33   .8087E-03    17    32   .1578E+03   .5324E-04    22   .8113E-03    22   .6642E-03
   280     .21182   -.04202     18    34   .7735E-03    17    32   .1534E+03   .5260E-04    22   .5184E-03    22   .9323E-03
   290     .21197   -.04203     18    34   .7349E-03    17    32   .1434E+03   .5151E-04    22   .4379E-03    21   .7850E-03
   300     .21212   -.04205     18    34   .7036E-03    17    32   .1358E+03   .5187E-04    22   .6149E-03    22   .7728E-03
   310     .21227   -.04208      8    63   .6727E-03    17    32   .1309E+03   .5470E-04    22   .3330E-03    21   .6594E-03
   320     .21243   -.04210      8    63   .6701E-03    17    32   .1253E+03   .5755E-04    22   [illegible]  21   [illegible]
   330     .21260   -.04212      8    63   .6693E-03    17    32   .1209E+03   .5968E-04    22   .2194E-03    21   .9638E-03
   340     .21278   -.04215      8    63   .6733E-03    17    32   .1151E+03   .6178E-04    65   .1903E-03    21   .3738E-03
   350     .21296   -.04218      8    63   .6725E-03    17    32   .1135E+03   .6371E-04    17   .1773E-03    21   .5448E-03
   360     .21314   -.04221      8    63   .6779E-03    17    32   .1064E+03   .6591E-04    17   .1013E-03    21   [illegible]
   370     .21333   -.04224      8    63   .6859E-03    17    32   .1026E+03   .6817E-04    17   .1369E-03    21   .4601E-03
   380     .21353   -.04227      8    53   .6955E-03    17    32   .9910E+02   .7194E-04    17   .1767E-03    21   .4337E-03
   390     .21374   -.04231     19    62   .5174E-03    17    32   .9550E+02   .7299E-04    17   .1859E-03    21   .4229E-03
   400     .21386   -.04232     16    61   .4732E-03    17    32   .9176E+02   .2942E-04    17   .1727E-03    21   .4146E-03
   410     .21387   -.04230      8     2   .4572E-03    17    32   .8812E+02   .6921E-05    64   [illegible]  21   .3760E-03
   420     .21380   -.04226      8     2   .4533E-03    17    32   .8445E+02   .3613E-04    64   .2939E-03    21   .3324E-03
   430     .21365   -.04222      8     2   .4496E-03    17    32   .8066E+02   .5669E-04    22   .3249E-03    21   .2442E-03
   440     .21345   -.04217      8     2   .4464E-03    17    32   .7689E+02   .7515E-04    22   [illegible]  21   .2639E-03
   450     .21322   -.04212      8     2   .4436E-03    17    32   .7312E+02   .8509E-04    22   .6294E-03    22   .1626E-03
   460     .21296   -.04207      8     2   .4412E-03    17    32   .6952E+02   [illegible]  22   [illegible]  63   .1530E-03
   470     .21270   -.04203      8     2   .4385E-03    17    32   .6674E+02   .9303E-04    22   [illegible]  65   .1372E-03
   480     .21243   -.04198      8     2   .4365E-03    17    32   .6327E+02   .9444E-04    22   .6608E-03    17   .1369E-03
   490     .21216   -.04193      8     2   .4329E-03    17    32   .6358E+02   .9582E-04    22   .6399E-03    17   .1540E-03
   500     .21188   -.04188      9     2   .4199E-03    17    32   .5767E+02   .9712E-04    22   [illegible]  17   [illegible]

                ******  ITERATION LIMIT REACHED  ******


                                                  ******************
                                                  *                *
                                                  *  FINAL OUTPUT  *
                                                  *                *
                                                  ******************

PRINTOUT IN  PHYSICAL  VARIABLES.

DEFINITION OF SIMILARITY PARAMETERS BY KRUPP

BOUNDARY CONDITION FOR FREE AIR

DIFFERENCE EQUATIONS ARE FULLY CONSERVATIVE.

KUTTA CONDITION IS ENFORCED.

             MACH =   1.1000000
            DELTA =    .0666667
            ALPHA =   2.0000000
                K =  -1.2456397

        PARAMETERS USED TO TRANSFORM VARIABLES
                 TO TRANSONIC SCALING

           CPFACT =    .1426087
           CDFACT =    .0089613
           CMFACT =    .1426087
           CLFACT =    .1426087
            YFACT =   2.4354913
            VFACT =   3.4377467
```

Figure 4.7.- Continued

```
                                   FINAL OUTPUT FOR FINE MESH

FORCE COEFFICIENTS, PRESSURE COEFFICIENT, AND MACH NUMBER
(OR SIMILARITY PARAMETER) ON BODY AND DIVIDING STREAM LINE.
                                                                               B = BEFORE OR BEHIND AIRFOIL
        CL =    .211883                                                        U = UPPER AIRFOIL SURFACE
        CM =  -.041882                                                         L = LOWER AIRFOIL SURFACE
       CP* =    .148116                                                        * = CRITICAL PRESSURE

                           LOWER                               UPPER
                           Y=0-                                Y=0+
 I        X          CP          M1                  CP          M1

 1   -4.026958   -0.000000    1.100000           -0.000000    1.100000
 2   -1.472160   -0.000000    1.100000           -0.000000    1.100000
 3   -1.170661     .000000    1.100000             .000000    1.100000
 4    -.972966     .000000    1.100000             .000000    1.100000
 5    -.869057     .009657    1.093759             .009657    1.093759
 6    -.653469     .165719     .987442             .165719     .987442
 7    -.470854     .335468     .856937             .335468     .856937
 8    -.295092     .391359     .830399             .391359     .830399
 9    -.170082     .458984     .768207             .458984     .768207
10    -.114702     .527590     .679689             .527590     .679689
11    -.075551     .597217     .602709             .597217     .602709
12    -.056199     .669409     .516768             .669409     .516768
13    -.032930     .749061     .392083             .749061     .392083
14    -.020510     .825603     .226750             .825603     .226750
15    -.011063     .911035    0.000000             .911035    0.000000
16    -.003465    1.009732    0.000000            1.009732    0.000000
                          AIRFOIL LEADING EDGE
17     .003024    1.209373    0.000000             .891538    0.000000
18     .008081     .851151     .272247             .451359     .755304
19     .014240     .614255     .582325             .259813     .917459
20     .019568     .502088     .705767             .154067     .993772
21     .025061     .426309     .776190             .073993    1.051294
22     .030961     .370353     .827542             .025077    1.083719
23     .037616     .325600     .865679            -.006739    1.095491
24     .045413     .280976     .896171            -.017892    1.111471
25     .054966     .251249     .924707            -.042969    1.127352
26     .067013     .216453     .950321            -.066554    1.142686
27     .082363     .181760     .975052            -.089946    1.156545
28     .101517     .146271    1.001306            -.113809    1.170961
29     .124281     .116795    1.026116            -.134861    1.183726
30     .149372     .080210    1.047033            -.154744    1.195574
31     .175006     .055675    1.063515            -.172279    1.209827
32     .202983     .034056    1.077363            -.187694    1.214832
33     .229170     .017100    1.088068            -.200988    1.222461
34     .255353     .002666    1.098668            -.211931    1.229015
35     .282029    -.011602    1.107948            -.221791    1.234694
36     .306213    -.022414    1.114351            -.233655    1.239625
37     .330950    -.032473    1.120732            -.238012    1.243960
38     .355299    -.041416    1.126375            -.244774    1.247615
39     .379323    -.049426    1.131485            -.251026    1.251249
40     .403081    -.056690    1.135927            -.256260    1.254335
41     .426629    -.063294    1.140075            -.261224    1.257126
42     .450015    -.069181    1.143719            -.265729    1.259664
43     .473283    -.074658    1.147106            -.269836    1.261985
44     .496460    -.079703    1.150219            -.273656    1.264147
45     .519562    -.084375    1.153095            -.277172    1.266087
46     .542609    -.088726    1.155767            -.280644    1.267918
47     .565740    -.092806    1.158267            -.283508    1.269637
48     .588807    -.096660    1.160623            -.286398    1.271243
49     .611881    -.100329    1.162862            -.289145    1.272776
50     .634971    -.103852    1.165007            -.291780    1.274241
51     .658084    -.107262    1.167081            -.294332    1.275659
52     .681229    -.110593    1.169103            -.296828    1.277046
53     .704417    -.113879    1.171093            -.299295    1.278415
54     .727663    -.117156    1.173071            -.301760    1.279781
55     .750988    -.120446    1.175056            -.304248    1.281158
56     .774418    -.123784    1.177074            -.306785    1.282561
57     .797986    -.127220    1.179141            -.309400    1.284006
58     .821742    -.130788    1.181284            -.312120    1.285507
59     .845736    -.134531    1.183528            -.314979    1.287082
60     .870039    -.138490    1.185902            -.318012    1.288752
61     .894736    -.142744    1.188439            -.321263    1.290539
62     .919931    -.147334    1.191172            -.324783    1.292471
63     .945757    -.152342    1.194149            -.328632    1.294581
64     .972377    -.157863    1.197422            -.332891    1.296912
65    1.000009    -.107273    1.167688            -.204055    1.224927
                          AIRFOIL TRAILING EDGE
66    1.020890    -.078062    1.149731            -.078062    1.149731
67    1.039441    -.051269    1.132599            -.051269    1.132599
68    1.092135    -.044428    1.128269            -.044428    1.128269
69    1.127723    -.041077    1.126665            -.041077    1.126665
70    1.167351    -.040240    1.125634            -.040240    1.125634
71    1.212946    -.038992    1.124848            -.038992    1.124848
72    1.268145    -.038174    1.124332            -.038174    1.124332
73    1.341106    -.038225    1.124365            -.038225    1.124365
74    1.456833    -.039577    1.125217            -.039577    1.125217
75    1.671514    -.038742    1.124691            -.038742    1.124691
76    1.876313    -.032662    1.120852            -.032662    1.120852
77    2.042961    -.023734    1.115191            -.023734    1.115191
78    2.209055    -.015180    1.109739            -.015180    1.109739
79    2.406774    -.011460    1.106721            -.011460    1.106721
80    2.702997    -.022465    1.114345            -.022465    1.114345
81    4.890373    -.048029    1.130529            -.048029    1.130529
```

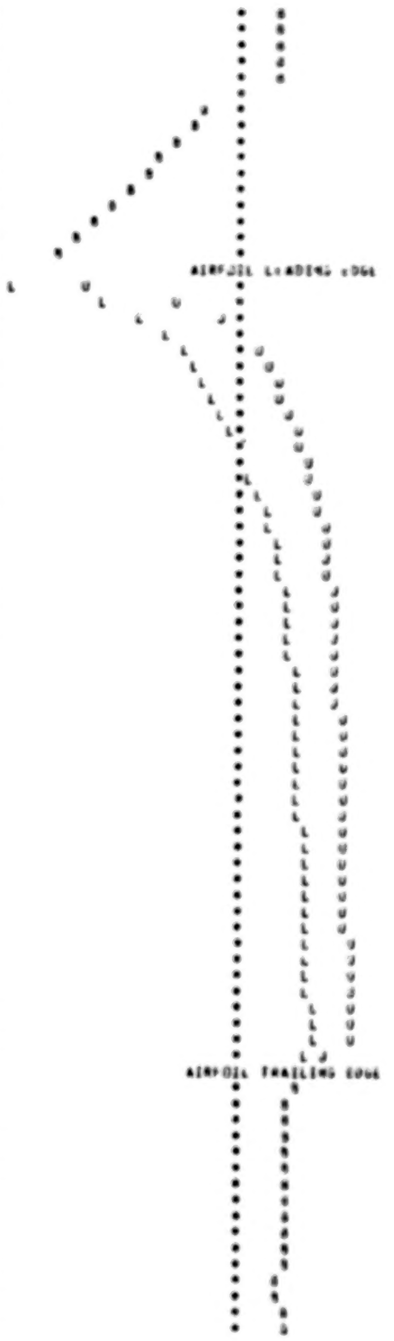


Figure 4.7.- Continued

```
   Y-GRID VALUES        Y(J) J=   1 TO  64
 -5.200000   -2.594100   -1.722837   -1.285224   -1.021060    -.843611    -.715781    -.618741
  -.542482    -.480484    -.429940    -.385439    -.347855    -.314984    -.285869    -.259787
  -.236188    -.214639    -.194799    -.176391    -.159191    -.143013    -.127702    -.113124
  -.099167    -.085732    -.072733    -.060093    -.047742    -.035615    -.023653    -.011800
   .011800     .023653     .035615     .047742     .060093     .072733     .085732     .099167
   .113124     .127702     .143013     .159191     .176391     .194799     .214639     .236188
   .259787     .285869     .314984     .347855     .385439     .429940     .480484     .542482
   .618741     .715781     .843611    1.021060    1.285224    1.722837    2.594100    5.200000
```

```
                                                    SONIC LINE OUTPUT
                                                    *****************
SONIC LINE COORDINATES

     Y          X-SONIC

   2.59410    -.27190      .49553
   1.72284    -.43613      .43866
   1.28522    -.52295      .35839
   1.02106    -.56660      .30646
    .84361    -.59380      .27041
    .71578    -.61232      .24028
    .61874    -.62348      .21731
    .54248    -.63493      .19866
    .48048    -.64203      .18360
    .42994    -.64744      .17036
    .38544    -.65163      .15428
    .34786    -.65496      .14896
    .31498    -.65758      .14051
    .28587    -.65969      .13261
    .25979    -.66143      .12487
    .23619    -.66287      .11832
    .21464    -.66409      .11214
    .19480    -.66512      .10597
    .17639    -.66601      .09993
    .15919    -.66678      .09471
    .14301    -.66744      .08941
    .12770    -.66803      .08378
    .11312    -.66854      .07868
    .09917    -.66900      .07364
    .08573    -.66940      .06861
    .07273    -.66975      .06281
    .06009    -.67006      .05705
    .04774    -.67034      .05109
    .03561    -.67058      .04429
    .02365    -.67079      .03677
    .01180    -.67097      .02738
BODY LOCATION
   -.01180    -.67125      .11132
   -.02365    -.67135      .12639
   -.03561    -.67142      .12962
   -.04774    -.67147      .13734
   -.06009    -.67149      .14460
   -.07273    -.67148      .15168
   -.08573    -.67144      .15940
   -.09917    -.67136      .16640
   -.11312    -.67124      .17292
   -.12770    -.67106      .18060
   -.14301    -.67087      .18730
   -.15919    -.67059      .19436
   -.17639    -.67025      .20119
   -.19480    -.66982      .20911
   -.21464    -.66929      .21730
   -.23619    -.66862      .22563
   -.25979    -.66786      .23383
   -.28587    -.66677      .24336
   -.31498    -.66547      .25386
   -.34786    -.66386      .26439
   -.38544    -.66165      .27640
   -.42994    -.65878      .29020
   -.48048    -.65487      .30519
   -.54248    -.64940      .32337
   -.61874    -.64169      .34473
   -.71578    -.63059      .37075
   -.84361    -.61410      .40182
  -1.02106    -.58911      .44388
  -1.28522    -.54893      .49611
  -1.72284    -.47480      .56551
  -2.59410    -.30048      .63446
```

Figure 4.7.- Continued

```
                                                   SHOCK WAVE DRAG AND TOTAL PRESSURE PROFILE OUTPUT
                                                   *************************************************

INVISCID WAKE PROFILES FOR INDIVIDUAL SHOCK WAVES WITHIN MOMENTUM CONTOUR

SHOCK  1
WAVE DRAG FOR THIS SHOCK=      .007130
      Y           CD(Y)        PO/POINF

 -2.59416019    .00161194    .99927451
 -1.72203680    .00085003    .99939063
 -1.20522355    .00138629    .99902612
 -1.02106025    .00162950    .99884618
  -.84361099    .00179065    .99871632
  -.71570104    .00195214    .99860295
  -.61874102    .00209996    .99849462
  -.54240166    .00223505    .99839717
  -.48048365    .00181152    .99870136
  -.42904031    .00190076    .99863739
  -.38543890    .00197252    .99858504
  -.34705516    .00203096    .99854405
  -.31498449    .00207920    .99852461
  -.28586858    .00211879    .99848128
  -.25970714    .00215193    .99845734
  -.23610771    .00217958    .99843756
  -.21463894    .00220274    .99842096
  -.19479857    .00222211    .99840721
  -.17639103    .00223829    .99839541
  -.15919133    .00225174    .99838577
  -.14301344    .00226283    .99837763
  -.12770160    .00227186    .99837135
  -.11312379    .00227907    .99836618
  -.09916674    .00228468    .99836216
  -.08573207    .00228882    .99835919
  -.07273326    .00229164    .99835717
  -.06004296    .00229324    .99835603
  -.04774152    .00229369    .99835570
  -.03561402    .00229305    .99835616
  -.02365311    .00229137    .99835736
  -.01179972    .00228868    .99835929
   .01179972    .00228027    .99836532
   .02365311    .00227454    .99836943
   .03561402    .00226775    .99837436
   .04774152    .00225986    .99837905
   .06004296    .00225086    .99838645
   .07273326    .00224056    .99839383
   .08573207    .00222884    .99840219
   .09916674    .00221572    .99841160
   .11312379    .00220096    .99842218
   .12770160    .00218439    .99843432
   .14301344    .00216578    .99844746
   .15919133    .00214485    .99846296
   .17639103    .00212127    .99847936
   .19479857    .00209464    .99849839
   .21463894    .00206445    .99852634
   .23610771    .00203007    .99854468
   .25970714    .00199072    .99857289
   .28586858    .00194546    .99860538
   .31498449    .00189283    .99864307
   .34705516    .00183133    .99868716
   .38543890    .00234772    .99831697
   .42904031    .00224426    .99839113
   .48048365    .00213317    .99847678
   .54240166    .00201386    .99855631
   .61874102    .00188953    .99864836
   .71570104    .00174673    .99874781
   .84361099    .00159398    .99885731
  1.02106025    .00141734    .99898396
  1.20522355    .00110417    .99919106
  1.72203680    .00080469    .99942313
  2.59416019    .00067453    .99951901

SHOCK WAVE EXTENDS OUTSIDE CONTOUR
PRINTOUT OF SHOCK LOSSES ARE NOT AVAILABLE FOR REST OF SHOCK


                                                                 DRAG COEFFICIENT OUTPUT
                                                                 ***********************

CALCULATION OF DRAG COEFFICIENT BY MOMENTUM INTEGRAL METHOD

BOUNDARIES OF CONTOUR USED                   CONTRIBUTION TO CD
UPSTREAM     X =    -1.170661                  CDUP    =     0.000000
DOWNSTREAM   X =      .727663                  CDDOWN  =      .011708
TOP          Y =     2.594100                  CUTOP   =      .034202
BOTTOM       Y =    -2.594100                  CUBOT   =      .035383
BODY AFT OF  X =      .727663                  CDBODY  =      .007620
                TOTAL CONTRIBUTIONS AROUND CONTOUR =          .027963

THERE ARE  1 SHOCKS INSIDE CONTOUR, TOTAL CDWAVE =            .007130

NOTE - ONE OR MORE SHOCKS EXTEND OUTSIDE OF CONTOUR
CDWAVE DOES NOT EQUAL TOTAL WAVE DRAG

DRAG CALCULATED FROM MOMENTUM INTEGRAL     CD      =          .035391


TIME TO RUN CASE WAS  32.44 SECONDS.
```

(b) Output.

Figure 4.7.- Concluded

```
SAMPLE CASE 5 - SUPERSONIC-NACA0006 AIRFOIL-FREE JET SIMULATION
$INP
 BCTYPE=3, H=2.0, EMACH=1.1, AMESH=.TRUE., ALPHA=2.0, DELTA=0.06,
 BCFOIL=1,
$END
```

(a) Input.

```
SAMPLE CASE 5 - SUPERSONIC - NACA0006 AIRFOIL - FREE JET SIMULATION


                              INPUT PARAMETERS
                              ****************

EMACH =  1.10000        POR =  0.00000        IMIN =    1     BCTYPE =  3         AMESH = T

DELTA =   .06000      CLSET =  0.00000       IMAXI =   81     BCFOIL =  1          PHYS = T

ALPHA =  2.00000        EPS =   .20000        JMIN =    1     PSTART =  1         PSAVE = F

   AK =  0.00000       RIGF =  0.00000       JMAXI =   64     PRTFLO =  1         KUTTA = T

  GAM =  1.40000      WCIRC =  1.00000       MAXIT =  300     IPRTER = 10           FCR = T

    F =  0.00000     CVERGE =   .00001          NU =  100     SIMDEF =  3

    H =  2.00000     DVERGE =     10.0          NL =   75       ICUT =  2

   WE = 1.80,1.90,1.95

  XIN
-4.020950    -1.472163    -1.170661     -.972466     -.809857     -.653469     -.474654    [illegible]
 -.170882     -.116742     -.079551     -.050198     -.032939     -.022518     -.011363    [illegible]
  .003024      .008653      .014240      .019588      .025361      .030961      .037014    [illegible]
  .054966      .067313      .082363      .101317      .124201      .149372      .175336    [illegible]
  .229179      .255353      .281029      .306213      .330990      .355299      .374023    [illegible]
  .426629      .453015      .478280      .498660      .519582      .542669      .565740    [illegible]
  .611881      .634971      .658784      .681629      .704417      .727663      .750488      .774610
  .797988      .821742      .845730      .873339      .894730      .919931      .945797      .972677
 1.000000     1.028049     1.059441     1.092135     1.127723     1.167351     1.212940    [illegible]
 1.341104     1.456833     1.671514     1.976513     2.342561     2.204055    [illegible]  [illegible]
 4.096373

  YIN
-1.000000     -.906455     -.821463     -.744089     -.673514     -.609036     -.553646     -.496010
 -.448463     -.402997     -.359253     -.320914     -.285702     -.253309     -.223696     -.196600
 -.171573     -.148707     -.128025     -.109135     -.092319     -.076544     -.062744     -.050426
 -.039566     -.030168     -.022364     -.015212     -.009701     -.005461     -.002413     -.000603
  .000603      .002413      .005461      .009701      .015212      .022364      .030168      .039566
  .050426      .062744      .076544      .092319      .109135      .128025      .148707      .171573
  .196600      .223696      .253309      .285702      .320914      .359253      .402997      .448463
  .496010      .553646      .609036      .673514      .744089      .821463      .906455     1.000000
```

Figure 4-8.- Sample test case 5 - supersonic, free jet simulation (abbreviated output): NACA 0006 airfoil, δ = 0.06, α = 2°, M_∞ = 1.1, H = 2.

```
                                          INTERMEDIATE OUTPUT FOR COARSE MESH

         WE =   1.0000     EPS =    .2000     MAXIT FOR THIS MESH =  125

ITERATION    CL          CM       IERR  JERR     ERROR      IRL   JRL        BIGRL          ERCIRC      ICPU     CPERRU       ICPL      CPLRRL

   10      .24775     -.05325      20     8    .4890E+00    19     9    .2022E+04     .6227E+00     15    .2482E+01     17    [illegible]
   20      .23063     -.04624      20     8    .3657E+00    19     9    .1346E+04     .4300E+00     17    .1172E+00     17    .2672E+00
   30      .21871     -.04396      20     8    .2199E+00    19     9    .1095E+04     .3237E+00     17    .2647E-01     17    .3594E-01
   40      .22180     -.04418      20     8    .2045E+00    19     9    .1115E+04     .3460E+00      5    .5007E-02     17    .4493E-01
   50      .22250     -.04427      20     8    .2061E+00    19     9    .1136E+04     .3472E+00      5    .3847E-02     17    .8072E-02
   60      .22250     -.04426      20     8    .2072E+00    19     9    .1140E+04     .3482E+00      5    .1760E-02      7    .1346E-02
   70      .22246     -.04424      20     8    .2063E+00    19     9    .1140E+04     .3483E+00      6    [illegible]     7    .5870E-03
   80      .22245     -.04424      20     8    .2058E+00    19     9    .1140E+04     .3481E+00      6    .4160E-03      7    .3416E-03
   90      .22244     -.04423      20     8    .2056E+00    19     9    .1142E+04     .3482E+00      5    [illegible]     7    .1743E-03
  100      .22244     -.04423      20     8    .2055E+00    19     9    .1140E+04     .3482E+00      5    [illegible]     7    .9822E-04
  110      .22244     -.04423      20     8    .2054E+00    19     9    .1140E+04     .3482E+00      6    .3469E-04      7    .4346E-04
  120      .22244     -.04423      20     8    .2054E+00    19     9    .1140E+04     .3482E+00      6    .2774E-04      7    .2230E-04

                     ******  ITERATION LIMIT REACHED  ******
```

```
                                          INTERMEDIATE OUTPUT FOR MEDIUM MESH

         WE =   1.9000     EPS =    .2000     MAXIT FOR THIS MESH =  250

ITERATION    CL          CM       IERR  JERR     ERROR      IRL   JRL        BIGRL          ERCIRC      ICPU     CPERRU       ICPL      CPLRRL

   10      .17667     -.03404      37    16    .3445E-01    35    16    .1868E+04     .3594E-01      4    .1298E+01      9    [illegible]
   20      .17601     -.03392      10    17    .3498E-02     9    16    .1790E+03     .1148E-03     33    .1347E-01     33    .3636E-01
   30      .17701     -.03391      10    17    .3036E-02     9    16    .1394E+03     .3108E-03     23    [illegible]    21    .8509E-02
   40      .17766     -.03374      10    17    .2310E-02     9    16    .1048E+03     .1822E-03     23    [illegible]    11    .7465E-02
   50      .17806     -.03357      10    17    .1736E-02     9    16    .7886E+02     .1528E-03     22    [illegible]    11    [illegible]
   60      .17844     -.03344      10    17    .1294E-02     9    16    .5884E+02     .1218E-03     22    [illegible]    11    [illegible]
   70      .17871     -.03334      10    17    .9553E-03     9    16    .4349E+02     .8333E-04     22    .2139E-02     11    .3731E-02
   80      .17896     -.03327      10    17    .7030E-03     9    16    .3204E+02     .5892E-04     22    [illegible]    11    .2772E-02
   90      .17904     -.03322      10    17    .5155E-03     9    16    .2357E+02     .4277E-04     22    .1117E-02     11    [illegible]
  100      .17914     -.03320      10    17    .3770E-03     9    16    .1722E+02     .3095E-04     22    .8113E-03     11    .1538E-02
  110      .17921     -.03316      10    17    .2752E-03     9    16    .1257E+02     .2247E-04     22    [illegible]    11    .1132E-02
  120      .17926     -.03314      10    17    .2036E-03     9    16    .9168E+01     .1632E-04     22    [illegible]    11    [illegible]
  130      .17930     -.03312      10    17    .1462E-03     9    16    .6680E+01     .1184E-04     22    .3115E-03     11    .6607E-03
  140      .17933     -.03311      10    17    .1064E-03     9    16    .4864E+01     .8588E-05     22    [illegible]    11    [illegible]
  150      .17935     -.03311      10    17    .7742E-04     9    16    .3540E+01     .6234E-05     22    [illegible]    11    [illegible]
  160      .17936     -.03310      10    17    .5632E-04     9    16    .2575E+01     .4529E-05     22    [illegible]    11    .2808E-03
  170      .17937     -.03310      10    17    .4096E-04     9    16    .1873E+01     .3294E-05     22    [illegible]    11    [illegible]
  180      .17938     -.03309      10    17    .2978E-04     9    16    .1362E+01     .2396E-05     22    .8314E-04     11    .1250E-03
  190      .17939     -.03309      10    17    .2165E-04     9    16    .9911E+00     .1743E-05     22    [illegible]    11    .9114E-04
  200      .17939     -.03309      10    17    .1573E-04     9    16    .7196E+00     .1268E-05     22    [illegible]    11    [illegible]
  210      .17940     -.03309      10    17    .1144E-04     9    16    .5230E+00     .9215E-06     22    [illegible]    11    [illegible]
  220      .17940     -.03309      10    17    .8310E-05     9    16    .3803E+00     .6697E-06     22    .1702E-04     11    [illegible]

                     ******  SOLUTION CONVERGED  ******
```

Figure 4-8.- Continued

```
                                   INTERMEDIATE OUTPUT FOR FINE MESH

        WE =   1.9500     EPS =     .2000     MAXIT FOR THIS MESH =  300

ITERATION     CL          CM      IERR  JERR     ERROR      IRL   JRL      BIGRL        ERCIRC     ICPU     CPLARU      ICPL     CPLARL

   10       .17616     -.03330     9    33    .3351E-02    16    32    .3015E+03    .2225E-04    17    [illegible]    17    [illegible]
   20       .17814     -.03325    15    18    .1169E-02    17    32    .2861E+03    .4672E-06    [illegible]    .1999E-02    21    [illegible]
   30       .17821     -.03319    17    36    .9756E-03    17    32    .2291E+03    .4689E-04    26    .2582E-02    21    .4969E-02
   40       .17843     -.03320    60    32    .7876E-03    17    32    .1934E+03    .6575E-04    65    .1524E-02    20    .2721E-02
   50       .17867     -.03322    18    35    .6715E-03    17    32    .1713E+03    .8253E-04    65    .1430E-02    22    .1394E-02
   60       .17886     -.03322    17    35    .5603E-03    17    32    .1480E+03    .5796E-04    36    [illegible]    22    .1970E-02
   70       [illegible]
   80       .17923     -.03318    17    36    .4137E-03    17    32    .1059E+03    .4173E-04    27    .3779E-03    21    .1764E-02
   90       .17924     -.03317    17    36    .3473E-03    17    32    .9046E+02    .3559E-04    27    [illegible]    [illegible]    [illegible]
  100       .17933     -.03316    17    36    .2928E-03    17    32    .7466E+02    .3017E-04    27    .5902E-03    21    [illegible]
  110       .17941     -.03315    17    36    .2472E-03    17    32    .6331E+02    .2511E-04    27    .5537E-03    21    .9955E-03
  120       .17947     -.03314    17    36    .2082E-03    17    32    .5514E+02    .2106E-04    27    .4273E-03    21    .9423E-03
  130       .17953     -.03313    17    36    .1755E-03    17    32    .4480E+02    .1762E-04    27    .3862E-03    21    .7144E-03
  140       .17957     -.03313    17    36    .1479E-03    17    32    .3776E+02    .1467E-04    27    [illegible]    21    .6311E-03
  150       .17961     -.03312    17    36    .1246E-03    17    32    .3162E+02    .1249E-04    [illegible]    .2574E-03    [illegible]    [illegible]
  160       .17964     -.03312    17    36    .1049E-03    17    32    .2681E+02    .1048E-04    [illegible]    [illegible]    21    [illegible]
  170       .17967     -.03311    17    36    .8838E-04    17    32    .2259E+02    .8823E-05    26    [illegible]    21    .8826E-03
  180       .17969     -.03311    17    36    .7446E-04    17    32    .1902E+02    .7546E-05    [illegible]    [illegible]    [illegible]    [illegible]
  190       .17971     -.03311    17    36    .6273E-04    17    32    .1603E+02    .6363E-05    [illegible]    [illegible]    21    [illegible]
  200       .17973     -.03311    17    36    .5284E-04    17    32    .1359E+02    .5339E-05    26    [illegible]    21    [illegible]
  210       .17974     -.03310    17    36    .4451E-04    17    32    .1136E+02    .4477E-05    [illegible]    [illegible]    [illegible]    [illegible]
  220       .17975     -.03311    17    36    .3754E-04    17    32    .9586E+01    .3756E-05    [illegible]    .7700E-04    21    .1541E-03
  230       .17976     -.03310    17    36    .3199E-04    17    32    .8077E+01    .3194E-05    [illegible]    .5574E-04    21    [illegible]
  240       .17977     -.03310    17    36    .2651E-04    17    32    .6804E+01    .2657E-05    26    [illegible]    [illegible]    [illegible]
  250       .17978     -.03310    17    36    .2247E-04    17    32    .5753E+01    .2236E-05    [illegible]    .4560E-04    21    .9211E-04
  260       .17978     -.03310    17    36    .1889E-04    17    32    .4829E+01    .1887E-05    26    [illegible]    21    .7764E-04
  270       .17979     -.03310    17    36    .1592E-04    17    32    .4069E+01    .1589E-05    26    .3313E-04    [illegible]    .6542E-04
  280       .17979     -.03309    17    36    .1342E-04    17    32    .3427E+01    .1338E-05    26    .2791E-04    21    .5514E-04
  290       .17979     -.03309    17    36    .1129E-04    17    32    .2887E+01    .1127E-05    26    .2352E-04    21    .4660E-04
  300       .17980     -.03309    17    36    .9511E-05    17    32    .2432E+01    .9468E-06    26    .1981E-04    21    .3914E-04

                     ******  SOLUTION CONVERGED  ******
```

```
                                                   ******************
                                                   *                *
                                                   *  FINAL OUTPUT  *
                                                   *                *
                                                   ******************

PRINTOUT IN  PHYSICAL  VARIABLES.

DEFINITION OF SIMILARITY PARAMETERS BY KRUPP

BOUNDARY CONDITION FOR FREE JET

DIFFERENCE EQUATIONS ARE FULLY CONSERVATIVE.

KUTTA CONDITION IS ENFORCED.

          MACH =   1.1000000
         DELTA =    .0600000
         ALPHA =   2.0000000
             K =  -1.2496397

      PARAMETERS USED TO TRANSFORM VARIABLES
              TO TRANSONIC SCALING

        CPFACT =    .1426867
        CDFACT =    .0085613
        CMFACT =    .1426867
        CLFACT =    .1426867
         YFACT =   2.4334913
         VFACT =   3.4377467
```

Figure 4-8.- Continued

```
                                   FINAL OUTPUT FOR FINE MESH

FORCE COEFFICIENTS, PRESSURE COEFFICIENT, AND MACH NUMBER
(OR SIMILARITY PARAMETER) ON BODY AND DIVIDING STREAM LINE.
                                                                               B = BEFORE OR BEHIND AIRFOIL
        CL =    .179790                                                        U = UPPER AIRFOIL SURFACE
        CM =  -.033093                                                         L = LOWER AIRFOIL SURFACE
       CP* =    .148116                                                        * = CRITICAL PRESSURE

                          LOWER                        UPPER
                          Y=0-                         Y=0+
 I        X          CP          M1              CP          M1

 1   -4.026950   -0.000000   1.100000       -0.000000    1.100000
 2   -1.472160   -0.000000   1.100000       -0.000000    1.100000
 3   -1.170661   -0.000000   1.100000       -0.000000    1.100000
 4    -.972966   -0.000000   1.100000       -0.000000    1.100000
 5    -.809057     .000000   1.100049         .000000    1.100049
 6    -.653469     .004128   1.097336         .004128    1.097336
 7    -.479854     .160203    .991394         .160203     .991394
 8    -.295092     .356466    .844091         .356466     .844091
 9    -.170002     .426632    .776067         .426632     .776067
10    -.114702     .505191    .732663         .505191     .732663
11    -.079551     .581962    .620416         .581962     .620416
12    -.050190     .664116    .518602         .664116     .518602
13    -.032932     .757466    .368031         .757466     .368031
14    -.020510     .873372   0.000000         .873372    0.000000
15    -.011063    1.034884   0.000000        1.034884    0.000000
16    -.003465    1.326956   0.000000        1.326956    0.000000
                         AIRFOIL LEADING EDGE
17     .003624    1.330277   0.000000         .926090    0.000000
18     .008863     .694742    .460798         .356148     .874264
19     .014243     .522670    .659413         .163714     .950601
20     .019560     .437645    .767794         .100125    1.033461
21     .025066     .362410    .818605         .060113    1.060552
22     .030961     .337291    .855477         .032970    1.080119
23     .037624     .290472    .886231         .004678    1.096081
24     .045613     .263379    .914046       -.019456     1.112497
25     .054966     .220482    .941305       -.043288     1.127352
26     .067013     .193695    .967500       -.067546     1.142723
27     .082363     .156126    .994305       -.092332     1.157976
28     .101517     .116170   1.022395       -.117120     1.173645
29     .124201     .078579   1.048136       -.140930     1.187355
30     .149372     .047527   1.068932       -.162869     1.200383
31     .179066     .020704   1.086523       -.182423     1.211075
32     .202503   -.002261    1.101456       -.199572     1.221064
33     .229170   -.022176    1.114207       -.214566     1.230534
34     .255351   -.039546    1.125193       -.227748     1.239105
35     .281029   -.054675    1.134814       -.239443     1.244793
36     .306213   -.068606    1.143850       -.249931     1.252742
37     .330950   -.082626    1.151034       -.259442     1.258123
38     .355299   -.092383    1.158008       -.268151     1.261126
39     .379323   -.102044    1.164394       -.276203     1.265545
40     .403081   -.112545    1.170285       -.283701     1.269738
41     .426620   -.121590    1.175756       -.290724     1.273653
42     .450015   -.130695    1.180868       -.297333     1.277326
43     .473280   -.138113    1.185672       -.303577     1.280787
44     .496463   -.145719    1.190211       -.309496     1.284699
45     .519582   -.152972    1.194523       -.315129     1.287165
46     .542669   -.159922    1.198641       -.323516     1.290125
47     .565740   -.166619    1.202595       -.329677     1.292904
48     .588807   -.173105    1.206412       -.333697     1.295694
49     .611881   -.179424    1.210119       -.339517     1.298345
50     .634971   -.185614    1.213740       -.342265     1.300935
51     .658004   -.191712    1.217297       -.344068     1.303484
52     .681229   -.197753    1.220816       -.349600     1.306012
53     .704417   -.203773    1.224249       -.354256     1.308537
54     .727663   -.209796    1.227703       -.358969     1.311077
55     .750900   -.215650    1.231270       -.363716     1.313649
56     .774410   -.221990    1.234803       -.368571     1.316267
57     .797900   -.228221    1.238376       -.373562     1.318954
58     .821742   -.234580    1.242014       -.378715     1.321721
59     .845736   -.241120    1.245738       -.384067     1.324509
60     .870630   -.247860    1.249571       -.389656     1.327376
61     .894736   -.254672    1.253539       -.395528     1.330728
62     .919931   -.262194    1.257673       -.401736     1.334611
63     .945757   -.269905    1.262012       -.408346     1.337919
64     .972377   -.278091    1.266402       -.415441     1.341276
65    1.000000   -.234962    1.242217       -.303553     1.280772
                         AIRFOIL TRAILING EDGE
66    1.020099   -.189936    1.216262       -.189936     1.216262
67    1.059441   -.188366    1.215343       -.188366     1.215343
68    1.092135   -.191968    1.217411       -.191968     1.217411
69    1.127723   -.196275    1.219951       -.196275     1.219951
70    1.167351   -.201266    1.222848       -.201266     1.222848
71    1.212946   -.207003    1.226169       -.207003     1.226169
72    1.260145   -.213038    1.230114       -.213038     1.230114
73    1.341104   -.222627    1.235165       -.222627     1.235165
74    1.456033   -.233196    1.241221       -.233196     1.241221
75    1.671514   -.237715    1.243799       -.237715     1.243799
76    1.876313   -.227569    1.238062       -.227569     1.238062
77    2.042961   -.200760    1.222599       -.200760     1.222599
78    2.209755   -.147135    1.191094       -.147135     1.191094
79    2.406774   -.016744    1.106902       -.016744     1.106902
80    2.701997     .095516   1.036618         .095516    1.036618
81    4.098373     .137149   1.028631         .137149    1.028631
```

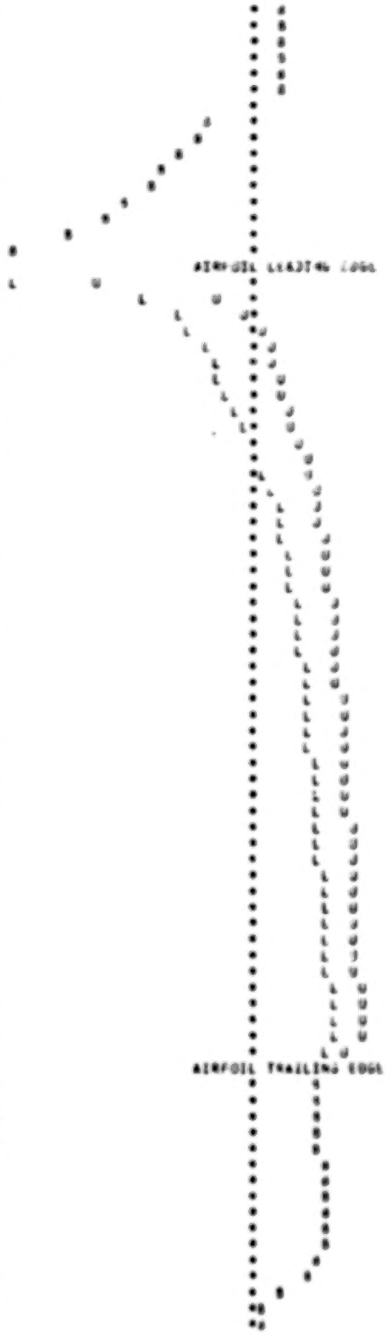


Figure 4-8.- Continued

```
***** CAUTION *****
MAXIMUM MACH NUMBER EXCEEDS 1.3
SHOCK JUMPS IN ERROR IF UPSTREAM NORMAL MACH NUMBER GREATER THAN 1.3


          Y-GRID VALUES      Y(J) J=   1 TO  64
        -2.000000   -1.812909   -1.642930   -1.488178   -1.347327   -1.218872   -1.100091    -.992019
         -.892925    -.801993    -.718565    -.641829    -.572404    -.508739    -.447391    -.392976
         -.342246    -.297595    -.256050    -.219269    -.184038    -.153167    -.125487    -.100852
         -.079132    -.060216    -.044067    -.030425    -.019401    -.011608    -.006027    -.001205
          .001205     .006027     .011608     .019401     .030425     .044067     .060216     .079132
          .100852     .125487     .153167     .184038     .219269     .256050     .297595     .342246
          .392976     .447391     .508739     .572404     .641829     .718565     .801993     .892925
          .992019    1.100091    1.218872    1.347327    1.488178    1.642930    1.812909    2.000000



                                                   SONIC LINE OUTPUT
                                                   *****************
SONIC LINE COORDINATES

     Y          X-SONIC

   1.40818   -.13936   -.06245
   1.34703   -.22076    .07896
   1.21087   -.25946    .07411
   1.10059   -.29580    .12619
    .99202   -.32988    .15905
    .89293   -.35538    .17072
    .80199   -.37752    .17473
    .71931   -.39604    .17586
    .64183   -.41325    .17391
    .57140   -.42762    .17122
    .50674   -.43996    .16467
    .44739   -.45044    .15576
    .39298   -.45923    .14623
    .34315   -.46649    .13689
    .29759   -.47241    .12761
    .25605   -.47714    .11783
    .21827   -.48100    .10839
    .18404   -.48425    .09894
    .15317   -.48675    .09039
    .12549   -.48866    .08137
    .10085   -.49009    .07344
    .07913   -.49114    .06562
    .06022   -.49189    .05720
    .04401   -.49242    .04965
    .03042   -.49278    .04219
    .01940   -.49302    .03503
    .01088   -.49316    .02837
    .00483   -.49324    .02229
    .00121   -.49329    .01756
 BODY LOCATION
   -.00121   -.49331    .08738
   -.00483   -.49335    .09073
   -.01088   -.49339    .09560
   -.01940   -.49343    .10152
   -.03042   -.49343    .10968
   -.04401   -.49336    .11789
   -.06022   -.49318    .12649
   -.07913   -.49283    .13660
   -.10085   -.49225    .14599
   -.12549   -.49135    .15642
   -.15317   -.49005    .16675
   -.18404   -.48823    .17699
   -.21827   -.48576    .18767
   -.25605   -.48247    .19783
   -.29759   -.47834    .20829
   -.34315   -.47335    .21825
   -.39298   -.46705    .22763
   -.44739   -.45923    .23562
   -.50674   -.44969    .24283
   -.57140   -.43823    .24803
   -.64183   -.42465    .25061
   -.71931   -.40872    .25099
   -.80199   -.39017    .24763
   -.89293   -.36854    .23656
   -.99202   -.34296    .21289
  -1.10059   -.31114    .17628
  -1.21087   -.26934    .10499
  -1.34703   -.22295    .02188
  -1.40818   -.19083   -.07634
```

Figure 4-8.- Continued

```
                                        SHOCK WAVE DRAG AND TOTAL PRESSURE PROFILE OUTPUT
                                        *************************************************

INVISCID WAKE PROFILES FOR INDIVIDUAL SHOCK WAVES WITHIN MOMENTUM CONTOUR

SHOCK  1
WAVE DRAG FOR THIS SHOCK=      .003082
       Y              CDSTY          PO/POINF

  -1.48817799      .00002214     .99998411
  -1.34702736      .00023112     .99983411
  -1.21807223      .00027830     .99980623
  -1.10039107      .00065035     .99952834
   -.99231012      .00062368     .99940052
   -.89292538      .00090406     .99929419
   -.80199344      .00114301     .99920030
   -.71836942      .00190623     .99906394
   -.64102866      .00147601     .99874123
   -.57140431      .00165076     .99881787
   -.50673795      .00185440     .99867062
   -.44739136      .00205564     .99851923
   -.39297572      .00224307     .99839615
   -.34314575      .00233036     .99816479
   -.29799466      .00279343     .99799741
   -.25624987      .00269874     .99844548
   -.21626493      .00222359     .99843596
   -.18403042      .00233062     .99832923
   -.15316763      .00241993     .99826542
   -.12946743      .00246056     .99821695
   -.10005215      .00294007     .99817072
   -.07913226      .00257097     .99815268
   -.06021992      .00260013     .99813652
   -.04486713      .00261319     .99812867
   -.03042464      .00261993     .99812234
   -.01946111      .00262001     .99812176
   -.01000238      .00261062     .99812271
   -.00462689      .00261045     .99812452
   -.00120127      .00261469     .99812934
    .00120127      .00261752     .99812657
    .00462689      .00261097     .99812823
    .01000238      .00260620     .99813161
    .01946111      .00259961     .99813791
    .03042464      .00259476     .99814708
    .04486713      .00256419     .99816178
    .06021992      .00255427     .99818823
    .07913226      .00249261     .99821316
    .10005215      .00245711     .99825249
    .12946743      .00236613     .99833977
    .15316763      .00227878     .99836439
    .18403042      .00217499     .99844197
    .21626493      .00205554     .99852643
    .25624987      .00272971     .99864312
    .29799466      .00244730     .99822049
    .34314575      .00224528     .99836733
    .39297572      .00202275     .99854276
    .44739136      .00189322     .99869500
    .50673795      .00165001     .99861715
    .57140431      .00146143     .99893806
    .64102866      .00132539     .99905608
    .71836942      .00117962     .99913566
    .80199344      .00103564     .99929741
    .89292538      .00089479     .99935434
    .99231012      .00074988     .99946238
   1.10039177      .00059797     .99957143
   1.21807223      .00046902     .99961715
   1.34702736      .00023669     .99983442
   1.48817799      .00002759     .99998125

SHOCK WAVE EXTENDS OUTSIDE CONTOUR
PRINTOUT OF SHOCK LOSSES ARE NOT AVAILABLE FOR REST OF SHOCK


                                        DRAG COEFFICIENT OUTPUT
                                        ***********************

CALCULATION OF DRAG COEFFICIENT BY MOMENTUM INTEGRAL METHOD

BOUNDARIES OF CONTOUR USED                 CONTRIBUTION TO CD
UPSTREAM    X =   -1.176661                 CDUP   =    0.000000
DOWNSTREAM  X =     .727663                 CDDOWN =     .117564
TOP         Y =    1.488178                 CDTOP  =     .002420
BOTTOM      Y =   -1.488178                 CDBOT  =     .113498
BODY AFT OF X =     .727663                 CDBODY =     .017922
             TOTAL CONTRIBUTIONS AROUND CONTOUR =     .134243

THERE ARE  1 SHOCKS INSIDE CONTOUR, TOTAL CDWAVE =     .108862

NOTE - ONE OR MORE SHOCKS EXTEND OUTSIDE OF CONTOUR
CDWAVE DOES NOT EQUAL TOTAL WAVE DRAG

DRAG CALCULATED FROM MOMENTUM INTEGRAL     CD     =     .039127


TIME TO RUN CASE WAS  26.11 SECONDS.
```

(b) Output.

Figure 4-6.- Concluded

```
SAMPLE CASE 6 - SUPERSONIC-NACA0006 AIRFOIL-POROUS TUNNEL WALL SIMULATION
$INP
 BCTYPE=5, POR=0.25, MAXIT=1000, AMESH=.TRUE., ALPHA=2.0, EMACH=1.1,
 H=2.0, DELTA=0.06, BCFOIL=1,
$END
```

(a) Input.

```
            SAMPLE CASE 6 - SUPERSONIC - NACA0006 AIRFOIL - POROUS TUNNEL WALL SIMULATION


                                       INPUT PARAMETERS
                                       ****************

   EMACH =  1.10000       POR =   .25000       IMIN =    1       BCTYPE =  5         AMESH = T

   DELTA =   .06000     CLSET =  0.00000      IMAXI =   81       BCFOIL =  1          PHYS = T

   ALPHA =  2.00000       EPS =   .20000       JMIN =    1       PSTART =  1         PSAVE = F

      AK =  0.00000      RIGF =  0.00000      JMAXI =   64       PRTFLO =  1         KUTTA = T

     GAM =  1.40000     WCIRC =  1.00000      MAXIT = 1000       IPRTER = 10           FCR = T

       F =  0.00000    CVERGE =   .00001         NU =  112       SIMDEF =  3

       H =  2.00000    DVERGE =     10.0         NL =   75         ICUT =  2

     WE = 1.80,1.90,1.95

    XIN
 -4.026950   -1.472180   -1.172661    -.972986    -.819257    -.653409    -.479926    -[illegible]
  -.170882    -.114712    -.075951    -.050146    -.032930    -.022516    -[illegible]    -[illegible]
   .003824     .008613     .014243     .019366     .025761     .036961     .037[illegible]     .045416
   .054966     .067217     .082369     .101947     .124051     .144972     .17[illegible]     .[illegible]
   .229179     .255288     .281124     .306213     .33[illegible]     .359[illegible]     .379323     .[illegible]
   .426829     .453015     .473208     .4999[illegible]     .519342     .542606     .565761     .5[illegible]
   .611881     .634971     .650284     .68122[illegible]     .704417     .727[illegible]     .75[illegible]     .77[illegible]
   .797986     .821742     .845736     .870[illegible]     .894735     .919931     .945[illegible]     .97[illegible]
  1.000000    1.028299    1.059441    1.092135    1.127723    1.167851    1.211946    1.2[illegible]
  1.343164    1.456033    1.671314    1.976313    2.342561    2.8[illegible]    [illegible]    [illegible]
  4.890373

    YIN
 -1.040016    -.966415    -.821469    -.744386    -.678514    -.609038    -.5[illegible]    -.4[illegible]
  -.446463    -.400907    -.359233    -.323416    -.285782    -.253360    -.22[illegible]    -.1[illegible]
  -.171573    -.148707    -.120425    -.100101    -.082C18    -.070504    -[illegible]    -[illegible]
  -.039368    -.03[illegible]    -.0[illegible]    -.019[illegible]    -.009701    -.005441    -[illegible]    -[illegible]
   .000003     .002411     .0[illegible]     .0[illegible]     .0[illegible]     .0[illegible]     [illegible]     [illegible]
   .050426     .067744     .07[illegible]     .092319     .1[illegible]     .1[illegible]     .1[illegible]     .1[illegible]
   .190488     .223696     .2[illegible]     .285732     .3[illegible]     .359253     .4[illegible]     .4[illegible]
   .446010     .750346     .6[illegible]     .673514     .744[illegible]     .822[illegible]     .9[illegible]     [illegible]


                              INTERMEDIATE OUTPUT FOR COARSE MESH

       WE =   1.8000     EPS =    .2000     MAXIT FOR THIS MESH =  250

ITERATION    CL          CM       IERR  JERR    ERROR      IRL   JRL      BIGRL         ERCIRC     ICPU    CPUMAX    ICPL    CPLMAX

    10     .24718    -.04579     21    16    .9428E-01     3     8     .1692E+03     .2613E-01    [illegible]  [illegible]  [illegible]  [illegible]
    20     .22445    -.04478     21     8    .1393E+[illegible]    19     8     .2696E+03     .7509E-01    [illegible]  [illegible]  [illegible]  [illegible]
    30     .21626    -.04569      5     9    .9930E-02     5     8     .4444E+02     .4663E-02    [illegible]  [illegible]  [illegible]  [illegible]
    40     .21560    -.04504      5     9    .3200E-02     5     8     .2443E+02     .1434E-03    [illegible]  [illegible]  [illegible]  [illegible]
    50     .21551    -.04588      5     9    [illegible]     5     8     [illegible]     .5266E-04    [illegible]  [illegible]  [illegible]  [illegible]
    60     .21546    -.04591      5     8    .1337E-02     5     8     .7522E+01     [illegible]    [illegible]  [illegible]  [illegible]  [illegible]
    70     .21546    -.04594      5     8    .5623E-03     5     8     .4199E+01     .3777E-05    [illegible]  [illegible]  [illegible]  [illegible]
    80     .21547    -.04595      5     8    .3113E-03     5     8     .2338E+01     [illegible]    [illegible]  [illegible]  [illegible]  [illegible]
    90     .21548    -.04596      5     8    .1747E-03     5     8     .1394E+01     [illegible]    [illegible]  [illegible]  [illegible]  [illegible]
   100     .21548    -.04597      5     8    .9745E-04     5     8     .7271E+00     [illegible]    [illegible]  [illegible]  [illegible]  [illegible]
   110     .21548    -.04597      5     8    .5436E-04     5     8     .4056E+00     [illegible]    [illegible]  [illegible]  [illegible]  [illegible]
   120     .21548    -.04597      5     8    .3032E-04     5     8     .2262E+00     [illegible]    [illegible]  [illegible]  [illegible]  [illegible]
   130     .21549    -.04597  [illegible]   8    .1692E-04     5     8     [illegible]     .1357E-06    [illegible]  [illegible]  [illegible]  [illegible]
   140     .21549    -.04597      5     8    .9438E-05     5     8     .7040E-01     [illegible]    [illegible]  [illegible]  [illegible]  [illegible]

                  ******  SOLUTION CONVERGED  ******
```

Figure 4-9.- Sample test case 6 - supersonic, perforated/porous wall simulation (abbreviated output): NACA 0006 airfoil, $\delta = 0.06$, $\alpha = 2°$, $M_\infty = 1.1$, $H = 2$.

```
                    INTERMEDIATE OUTPUT FOR MEDIUM MESH

      WE =  1.9000     EPS =    .2000     MAXIT FOR THIS MESH = [illegible]

ITERATION    CL          CM      IERR  JERR    ERROR     IRL   JRL    BIGRL    [illegible]

    10     .22231     -.04389    [illegible]
    20     .21429     -.04389    [illegible]
    30     .21337     -.04365    [illegible]
    40     .21396     -.04369    [illegible]
    50     .21405     -.04372    [illegible]
    60     .21413     -.04376    [illegible]
    70     .21416     -.04376    [illegible]
    80     .21419     -.04378    [illegible]
    90     .21420     -.04378    [illegible]
   100     .21421     -.04381    [illegible]
   110     .21422     -.04382    [illegible]
   120     .21423     -.04383    [illegible]
   130     .21423     -.04384    [illegible]
   140     .21424     -.04384    [illegible]
   150     .21424     -.04385    [illegible]
   160     .21424     -.04386    [illegible]
   170     .21424     -.04386    [illegible]
   180     .21423     -.04386    [illegible]
   190     .21423     -.04387    [illegible]
   200     .21425     -.04387    [illegible]
   210     .21425     -.04387    [illegible]
   220     .21425     -.04388    [illegible]
   230     .21426     -.04388    [illegible]
   240     .21426     -.04388    [illegible]
   250     .21426     -.04388    [illegible]
   260     .21426     -.04388    [illegible]
   270     .21426     -.04388    [illegible]
   280     .21426     -.04389    [illegible]
   290     .21426     -.04389    [illegible]
   300     .21426     -.04389    [illegible]
   310     .21426     -.04389    [illegible]
   320     .21426     -.04389    [illegible]
   330     .21426     -.04389    [illegible]
   340     .21426     -.04389    [illegible]
   350     .21426     -.04389    [illegible]
   360     .21426     -.04389    [illegible]

          ******  SOLUTION CONVERGED  ******
```

```
                    INTERMEDIATE OUTPUT FOR FINE MESH

      WE =  1.9500     EPS =    .2000     MAXIT FOR THIS MESH = [illegible]

ITERATION    CL          CM      IERR  JERR    ERROR     IRL   JRL    BIGRL    [illegible]

    10     .21464     -.04461    [illegible]
    20     .21513     -.04427    [illegible]
    30     .21533     -.04431    [illegible]
    40     .21544     -.04429    [illegible]
    50     .21551     -.04426    [illegible]
    60     .21556     -.04423    [illegible]
    70     .21560     -.04421    [illegible]
    80     .21564     -.04419    [illegible]
    90     .21567     -.04417    [illegible]
   100     .21569     -.04415    [illegible]
   110     .21571     -.04414    [illegible]
   120     .21573     -.04413    [illegible]
   130     .21574     -.04411    [illegible]
```

Figure 6-9.- Continued

```
****** ITERATION LIMIT REACHED ******
```

Figure 4-9.- Continued

TABLE OF CONTENTS

Section — Page No.

1. SUMMARY 1 1/A7

2. INTRODUCTION 2 1/A8

3. LIST OF SYMBOLS 5 1/A11

4. GENERAL USER'S MANUAL 8 1/A14

4.1 Description of Program 8 1/A14

4.1.1 Differential equations and transonic scaling 8 1/A14
4.1.2 Geometry 9 1/B1
4.1.3 Boundary conditions 9 1/B1
4.1.4 Aerodynamic variables 13 1/B5
4.1.5 Numerical formulation 13 1/B5
4.1.6 Operating procedure 17 1/B9
4.1.7 Overall program flow chart 19 1/B11

4.2 Program Input 20 1/B12

4.2.1 Dictionary of input variables 20 1/B12
4.2.2 Control options 22 1/B14
4.2.3 Default values 30 1/C8
4.2.4 Input data format 31 1/C9

4.3 Error and Information Messages 34 1/C12

4.4 Program Output 38 1/D2

4.4.1 General description of the output 38 1/D2
4.4.2 Sample cases 41 1/D5

4.5 Comparisons with Data and Other Methods 42 1/D6

4.6 Program Run Times 44 1/D8

4.7 Machine Compatability 46 1/D10

5. PROGRAMMER'S MANUAL 102 2/A11

5.1 Dictionary of Subroutine Variables in Common 102 2/A11

5.2 Description of Individual Subroutines 116 2/B11

5.2.1 Subroutine ANGLE 116 2/B11
⋮ ⋮
5.2.53 Subroutine VROOTS 145 2/E1

APPENDIX - LISTING OF COMPUTER PROGRAM TSFOIL 146 2/E2

REFERENCES 177 3/B11

```
                                         ******************
                                         *                *
                                         *  FINAL OUTPUT  *
                                         *                *
                                         ******************

PRINTOUT IN  PHYSICAL  VARIABLES.

DEFINITION OF SIMILARITY PARAMETERS BY KRUPP

BOUNDARY CONDITION FOR POROUS WALL

DIFFERENCE EQUATIONS ARE FULLY CONSERVATIVE.

KUTTA CONDITION IS ENFORCED.

              MACH =   1.1000000
             DELTA =    .0600000
             ALPHA =   2.0000000
                 K =  -1.2456397

         PARAMETERS USED TO TRANSFORM VARIABLES
                TO TRANSONIC SCALING

            CPFACT =    .1426887
            CDFACT =    .0085613
            CMFACT =    .1426887
            CLFACT =    .1426887
             YFACT =   2.4354913
             VFACT =   3.4377467
```

Figure 4-9.- Continued

Microfilmed From Best Available Copy

```
                                                  FINAL OUTPUT FOR FINE MESH

FORCE COEFFICIENTS, PRESSURE COEFFICIENT, AND MACH NUMBER
(OR SIMILARITY PARAMETER) ON BODY AND DIVIDING STREAM LINE.

        CL =   .215695
        CM =  -.043689
       CP* =   .148116

                              LOWER                          UPPER
                              Y=0-                           Y=0+
  I       X           CP          M1               CP           M1

  1   -4.826958   -0.000000    1.130000        -0.000000    1.130000
  2   -1.472160   -0.000000    1.130000        -0.000000    1.130000
  3   -1.170661     .000000    1.130000          .000000    1.130000
  4    -.972966     .000000    1.130000          .000000    1.130000
  5    -.809857     .000000    1.130000          .000000    1.130000
  6    -.653469     .137155    1.037740          .137155    1.037740
  7    -.479854     .314178     .874388          .314178     .874388
  8    -.295092     .386917     .813280          .386917     .813280
  9    -.170882     .454829     .751757          .454829     .751757
 10    -.114702     .525168     .682273          .525168     .682273
 11    -.075551     .597971     .601822          .597971     .601822
 12    -.050195     .677463     .499457          .677463     .499457
 13    -.032930     .768879     .346229          .768879     .346229
 14    -.020518     .883169    0.000000          .883169    0.000000
 15    -.011063    1.048274    0.000000         1.048274    0.000000
 16    -.003465    1.332789    0.000000         1.332789    0.000000
                         AIRFOIL LEADING EDGE
 17     .003024    1.350567    0.000000          .913249    0.000000
 18     .008363     .727384     .422738          .294986     .893329
 19     .014240     .549219     .656743          .152382     .996971
 20     .019568     .454576     .742528          .091298    1.039498
 21     .025061     .408406     .794328          .053712    1.064822
 22     .030961     .364456     .832633          .025655    1.083229
 23     .037614     .326508     .864334          .001710    1.098897
 24     .045413     .291394     .892669        -.020795    1.113321
 25     .054986     .257163     .919452        -.042874    1.127292
 26     .067013     .222728     .945629        -.065286    1.141299
 27     .082363     .187051     .971423        -.088190    1.155438
 28     .101517     .152341     .997003        -.110994    1.169345
 29     .124201     .116842    1.021930        -.132743    1.182437
 30     .149372     .085611    1.043369        -.152511    1.194255
 31     .175806     .061522    1.060279        -.173003    1.204588
 32     .202583     .039583    1.074188        -.185195    1.213493
 33     .229179     .021680    1.085867        -.199299    1.221127
 34     .255393     .006706    1.095635        -.209535    1.227690
 35     .281029    -.006289    1.104045        -.219493    1.233284
 36     .306213    -.017666    1.111325        -.228136    1.239523
 37     .330953    -.027676    1.117694        -.235756    1.242682
 38     .355299    -.036595    1.123311        -.242541    1.246545
 39     .379323    -.044682    1.128303        -.248618    1.249997
 40     .403081    -.051666    1.132765        -.254094    1.253099
 41     .426629    -.058026    1.136781        -.259356    1.255903
 42     .450015    -.063857    1.140411        -.263575    1.258451
 43     .473283    -.069173    1.143711        -.267717    1.260774
 44     .496463    -.074048    1.146727        -.271514    1.262915
 45     .519582    -.078534    1.149499        -.275031    1.264888
 46     .542669    -.082092    1.152060        -.278351    1.266719
 47     .565740    -.086565    1.154441        -.281358    1.268429
 48     .588807    -.090196    1.156668        -.284236    1.270036
 49     .611881    -.093622    1.158766        -.286963    1.271558
 50     .634971    -.096881    1.160758        -.289569    1.273017
 51     .658084    -.100006    1.162665        -.292080    1.274427
 52     .681229    -.103032    1.164509        -.294522    1.275785
 53     .704417    -.105993    1.166309        -.296925    1.277097
 54     .727663    -.108921    1.168088        -.299320    1.278417
 55     .750988    -.111852    1.169865        -.301687    1.279767
 56     .774418    -.114821    1.171663        -.304109    1.281081
 57     .797988    -.117885    1.173526        -.306594    1.282436
 58     .821742    -.121024    1.175413        -.309171    1.283879
 59     .845736    -.124339    1.177408        -.311874    1.285371
 60     .870039    -.127857    1.179524        -.314740    1.286951
 61     .894736    -.131632    1.181791        -.317811    1.288641
 62     .919931    -.135725    1.184244        -.321138    1.290474
 63     .945757    -.140212    1.186927        -.324793    1.292471
 64     .972377    -.145186    1.189893        -.328821    1.294684
 65    1.000000    -.097686    1.160883        -.188877    1.213645
                         AIRFOIL TRAILING EDGE
 66    1.020899    -.043830    1.127893        -.043830    1.127893
 67    1.059441    -.033456    1.121357        -.033456    1.121357
 68    1.092135    -.029416    1.118797        -.029416    1.118797
 69    1.127723    -.026085    1.116684        -.026085    1.116684
 70    1.167351    -.022915    1.114670        -.022915    1.114670
 71    1.212946    -.019723    1.112638        -.019723    1.112638
 72    1.268145    -.016427    1.110536        -.016427    1.110536
 73    1.341184    -.013170    1.108455        -.013177    1.108455
 74    1.456833    -.010805    1.106942        -.010805    1.106942
 75    1.671514    -.008974    1.105768        -.008974    1.105768
 76    1.876313    -.007730    1.104970        -.007730    1.104970
 77    2.042561    -.009657    1.106206        -.009657    1.106206
 78    2.209055    -.017748    1.111379        -.017748    1.111379
 79    2.406774    -.034402    1.121952        -.034402    1.121952
 80    2.702997    -.043931    1.127956        -.043931    1.127956
 81    4.890373    -.041606    1.126494        -.041606    1.126494
```

```
B = BEFORE OR BEHIND AIRFOIL
U = UPPER AIRFOIL SURFACE
L = LOWER AIRFOIL SURFACE
* = CRITICAL PRESSURE
```

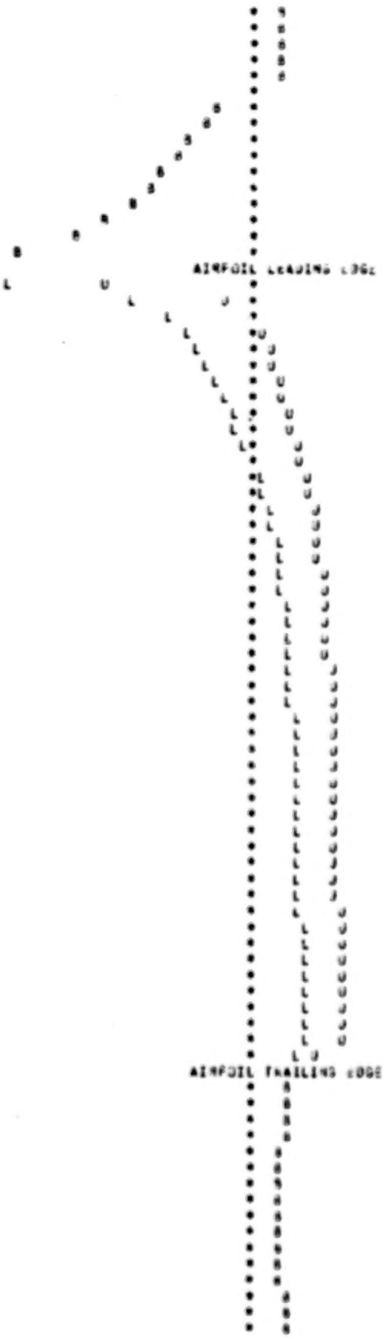


Figure 4-9.- Continued

Microfilmed From
Best Available Copy

```
      Y-GRID VALUES      Y(J) J=   1 TO  64
  -2.000000   -1.812909   -1.642933   -1.488178   -1.347027   -1.218072   -1.100091    -.992019
   -.892925    -.801993    -.718505    -.641829    -.571404    -.506738    -.447391    -.392976
   -.343146    -.297595    -.256050    -.218269    -.184038    -.153167    -.125487    -.100852
   -.079132    -.060216    -.044007    -.030425    -.019401    -.010882    -.004827    -.001205
    .001205     .004827     .010882     .019401     .030425     .044007     .060216     .079132
    .100852     .125487     .153167     .184038     .218269     .256050     .297595     .343146
    .392976     .447391     .506738     .571404     .641829     .718505     .801993     .892925
    .992019    1.100091    1.218072    1.347027    1.488178    1.642930    1.812909    2.000000
```

```
                                              SONIC LINE OUTPUT
                                              *****************
 SONIC LINE COORDINATES

      Y           X-SONIC

   2.00000     -.33996      .57889
   1.81291     -.38427      .52967
   1.64293     -.42050      .48093
   1.48818     -.45417      .43681
   1.34703     -.48533      .39794
   1.21807     -.50724      .36341
   1.10009     -.52586      .33294
    .99202     -.54224      .30665
    .89293     -.55682      .28327
    .80199     -.56980      .26189
    .71851     -.58131      .24204
    .64183     -.59143      .22341
    .57140     -.60026      .20628
    .50674     -.60789      .19055
    .44739     -.61441      .17532
    .39298     -.61993      .16192
    .34315     -.62455      .14836
    .29759     -.62839      .13645
    .25605     -.63155      .12429
    .21827     -.63411      .11383
    .18404     -.63617      .10299
    .15317     -.63781      .09239
    .12549     -.63910      .08355
    .10085     -.64010      .07490
    .07913     -.64086      .06589
    .06022     -.64144      .05780
    .04401     -.64187      .04983
    .03042     -.64218      .04207
    .01940     -.64240      .03470
    .01088     -.64255      .02781
    .00483     -.64265      .02163
    .00121     -.64270      .01685
 BODY LOCATION
   -.00121     -.64274      .10945
   -.00483     -.64279      .10899
   -.01088     -.64286      .11434
   -.01940     -.64295      .12085
   -.03042     -.64304      .12907
   -.04401     -.64312      .13853
   -.06022     -.64315      .14789
   -.07913     -.64310      .15936
   -.10085     -.64295      .17058
   -.12549     -.64265      .18276
   -.15317     -.64214      .19541
   -.18404     -.64137      .20896
   -.21827     -.64026      .22239
   -.25605     -.63873      .23687
   -.29759     -.63671      .25172
   -.34315     -.63408      .26775
   -.39298     -.63073      .28396
   -.44739     -.62656      .30142
   -.50674     -.62142      .32005
   -.57140     -.61521      .33958
   -.64183     -.60780      .36022
   -.71851     -.59906      .38225
   -.80199     -.58889      .40608
   -.89293     -.57719      .43225
   -.99202     -.56385      .46132
  -1.10009     -.54874      .49293
  -1.21807     -.53167      .52827
  -1.34703     -.51218      .56770
  -1.48818     -.48915      .61174
  -1.64293     -.45651      .65995
  -1.81291     -.41901      .71061
  -2.00000     -.37572      .78136
```

Figure 4-9.- Continued

```
                                                 SHOCK WAVE DRAG AND TOTAL PRESSURE PROFILE OUTPUT
                                                 *************************************************

INVISCID WAKE PROFILES FOR INDIVIDUAL SHOCK WAVES WITHIN MOMENTUM CONTOUR

SHOCK  1
WAVE DRAG FOR THIS SHOCK=     .006437
      Y           CD(Y)        PO/POINF

 -1.81290940    .00074477    .99946629
 -1.64293038    .00077644    .99944339
 -1.48817799    .00104243    .9992527C
 -1.34702736    .00120194    .99913836
 -1.21007223    .00134241    .99903765
 -1.10009107    .00147094    .99894551
  -.99201912    .00159340    .99885773
  -.89292538    .00171416    .99877115
  -.80199344    .00183630    .9986836C
  -.71856942    .00196176    .99859365
  -.64182064    .00209155    .99850001
  -.57140431    .00222586    .99840437
  -.50673795    .00236387    .99830539
  -.44739136    .00250436    .99820468
  -.39297572    .0026452C    .99810371
  -.34314575    .00278373    .99800440
  -.29759466    .00291608    .99790895
  -.25604907    .00304137    .99781976
  -.21826935    .00315408    .99773891
  -.18403842    .00325233    .99766847
  -.15316703    .00333425    .99760971
  -.12540743    .00339920    .99756319
  -.10085215    .00344746    .99752850
  -.07913226    .00348062    .99750482
  -.06021992    .00350102    .99749019
  -.04400713    .00351149    .99748268
  -.03042464    .00351495    .9974802C
  -.01940111    .00351407    .99748083
  -.01088238    .0035111C    .99748297
  -.00482609    .00350777    .99748535
  -.00120527    .00350529    .99748713
   .00120527    .00350345    .99748845
   .00482609    .00350030    .99749065
   .01088238    .00349447    .99749480
   .0194C111    .00348451    .99750202
   .03042464    .00346807    .99751324
   .04400713    .00344954    .99752497
   .06021992    .00341232    .99755377
   .07913226    .00336708    .99758621
   .10085215    .00330002    .99762855
   .12540743    .00323402    .99768160
   .15316703    .00314487    .99774551
   .18403842    .00304139    .99781969
   .21826935    .00292931    .9979C290
   .25604907    .00279911    .99799338
   .29759466    .00266567    .99808903
   .34314575    .00252002    .99818772
   .39297572    .00238097    .99828739
   .44739136    .00225094    .99838631
   .50673795    .00211595    .99848312
   .57140431    .00198513    .99857690
   .64182064    .00185909    .99866726
   .7185C942    .00173771    .99875427
   .80199344    .00162012    .99883857
   .89292538    .00150465    .99892135
   .99201912    .00138075    .99900443
  1.10009107    .0012609C    .99909035
  1.21007223    .00114057    .99918235
  1.34702736    .00099855    .9992841C
  1.48817799    .00074000    .99940945
  1.64293038    .00072927    .99947735
  1.8129C940    .00067902    .99951322

SHOCK WAVE EXTENDS OUTSIDE CONTOUR
PRINTOUT OF SHOCK LOSSES ARE NOT AVAILABLE FOR REST OF SHOCK

                                                            DRAG COEFFICIENT OUTPUT
                                                            ***********************

CALCULATION OF DRAG COEFFICIENT BY MOMENTUM INTEGRAL METHOD

BOUNDARIES OF CONTOUR USED                     CONTRIBUTION TO CD
UPSTREAM    X =   -1.17C661                     CDUP    =   0.003030
DOWNSTREAM  X =     .727663                     CDDOWN  =    .039068
TOP         Y =    1.812909                     CDTOP   =    .005369
BOTTOM      Y =   -1.812909                     CDBOT   =    .007056
BODY AFT OF X =     .727663                     CDBODY  =    .007856
                TOTAL CONTRIBUTIONS AROUND CONTOUR =         .029040

THERE ARE  1 SHOCKS INSIDE CONTOUR, TOTAL CDWAVE =           .006437

NOTE - ONE OR MORE SHOCKS EXTEND OUTSIDE OF CONTOUR
CDWAVE DOES NOT EQUAL TOTAL WAVE DRAG

DRAG CALCULATED FROM MOMENTUM INTEGRAL    CD     =           .035405


TIME TO RUN CASE WAS  62.07 SECONDS.
```

(b) Output.

Figure 4-9.- Concluded

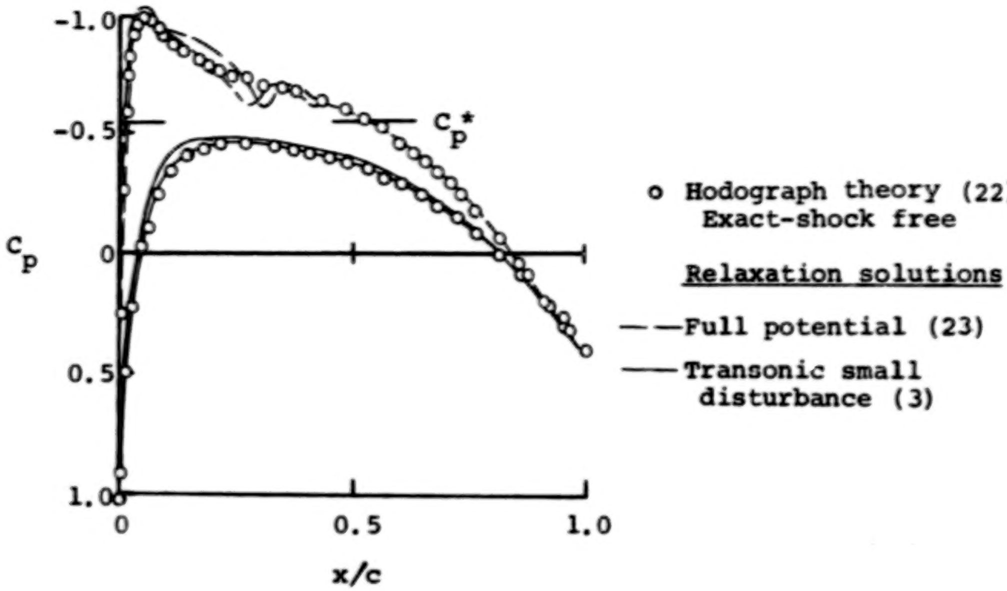


Figure 4-10.- Comparison of hodograph and relaxation solutions for the surface pressure distribution on a NLR quasi-elliptical airfoil with $\delta = 0.1212$, $\alpha = 1.32^\circ$ and $M_\infty = 0.7557$.

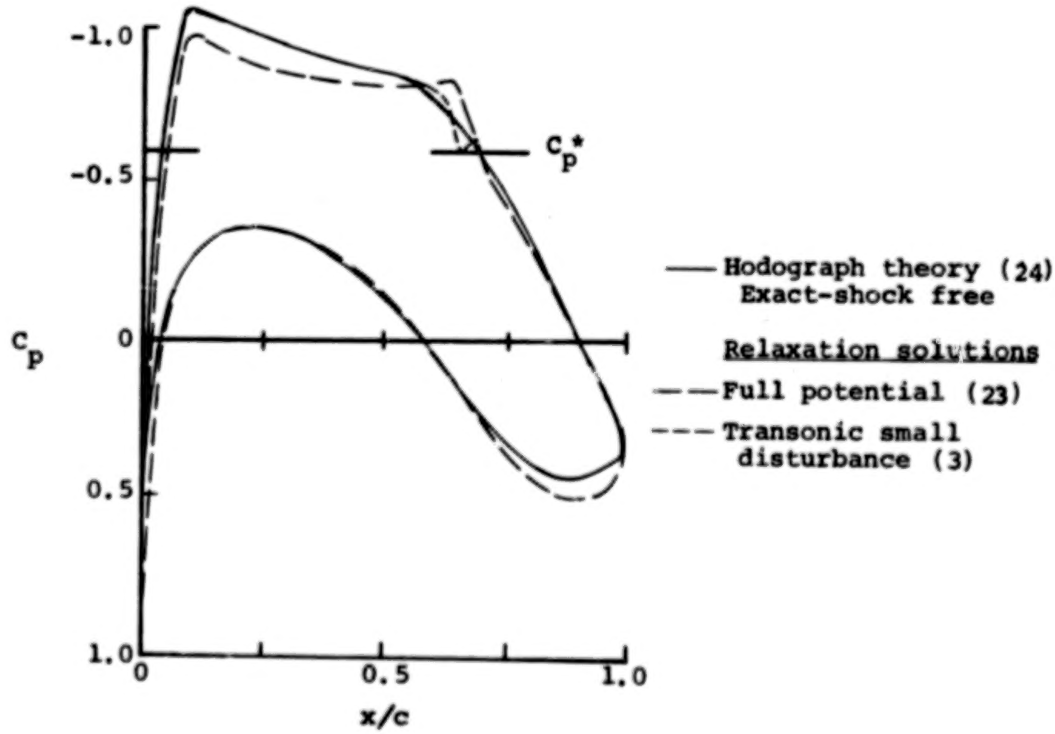


Figure 4-11.- Comparison of hodograph and relaxation solutions for the surface pressure distribution on a Korn airfoil with $\delta = 0.115$, $\alpha = 0.12^\circ$, and $M_\infty = 0.75$.

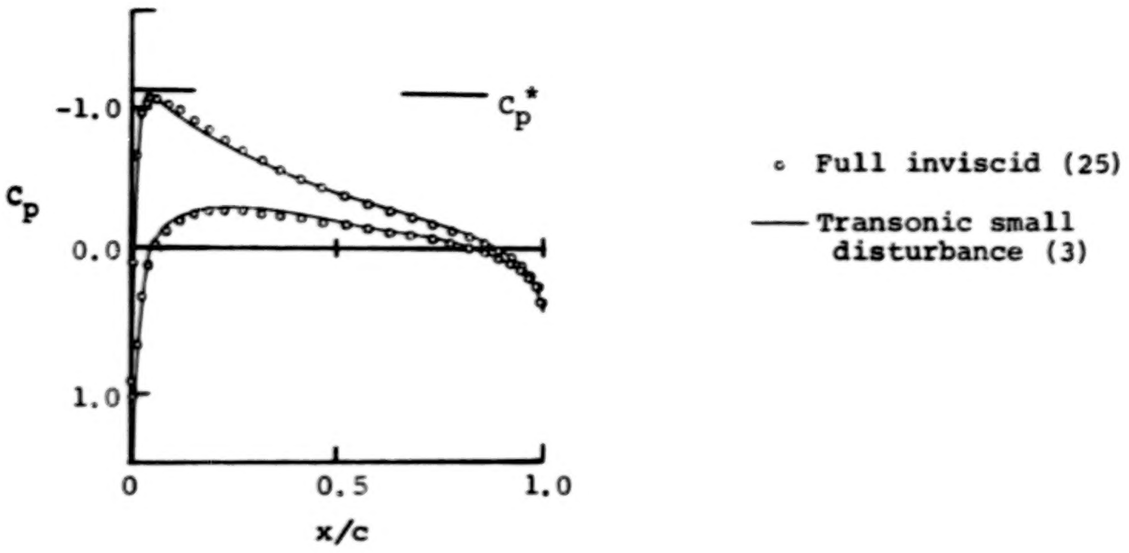


Figure 4-12.- Comparison of relaxation solutions for the surface pressure distribution of a NACA 0012 airfoil at $\alpha = 2^\circ$, $M_\infty = 0.630$.

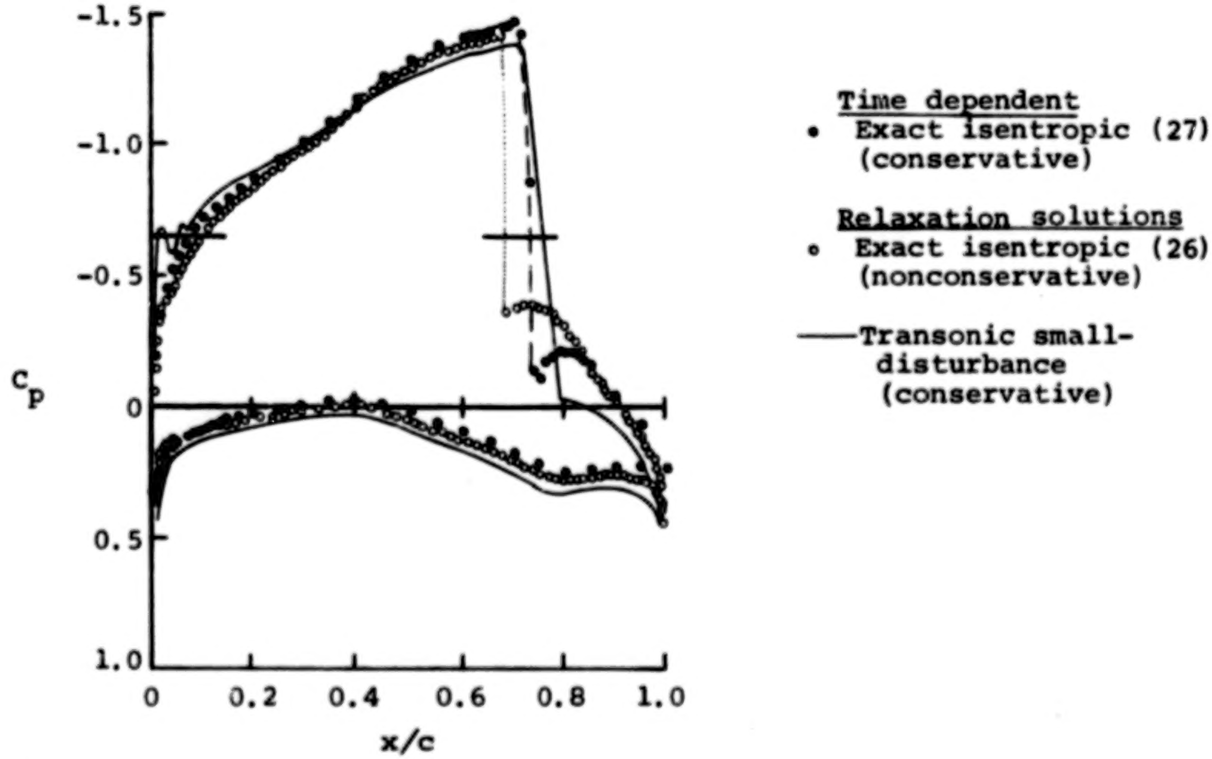


Figure 4-13.- Comparison of conservative time-dependent and nonconservative relaxation solutions of exact isentropic equation and the small-disturbance conservative solution for the surface pressure distribution on an NACA 64A410 airfoil at $M_\infty = 0.735$ and $\alpha = 1^\circ$.

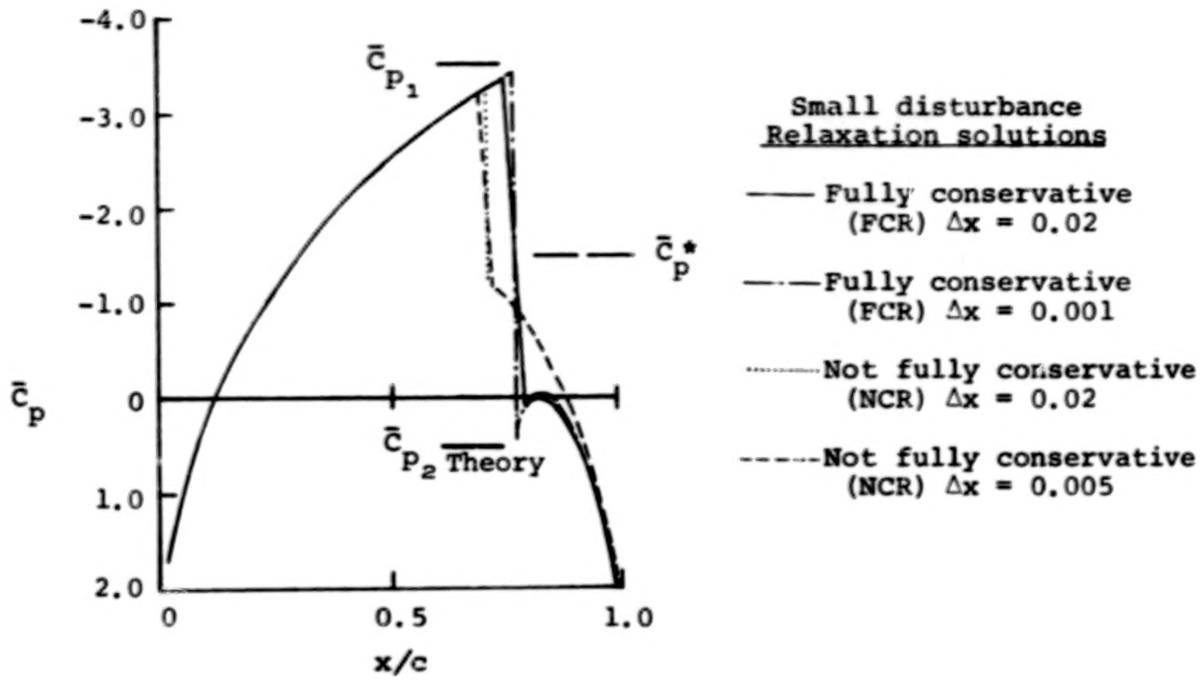


Figure 4-14.- Comparison of FCR and NCR solutions for the similarity surface pressure distribution on a parabolic arc airfoil with $\delta = 0.06$ and $M_\infty = 0.872$. (19)

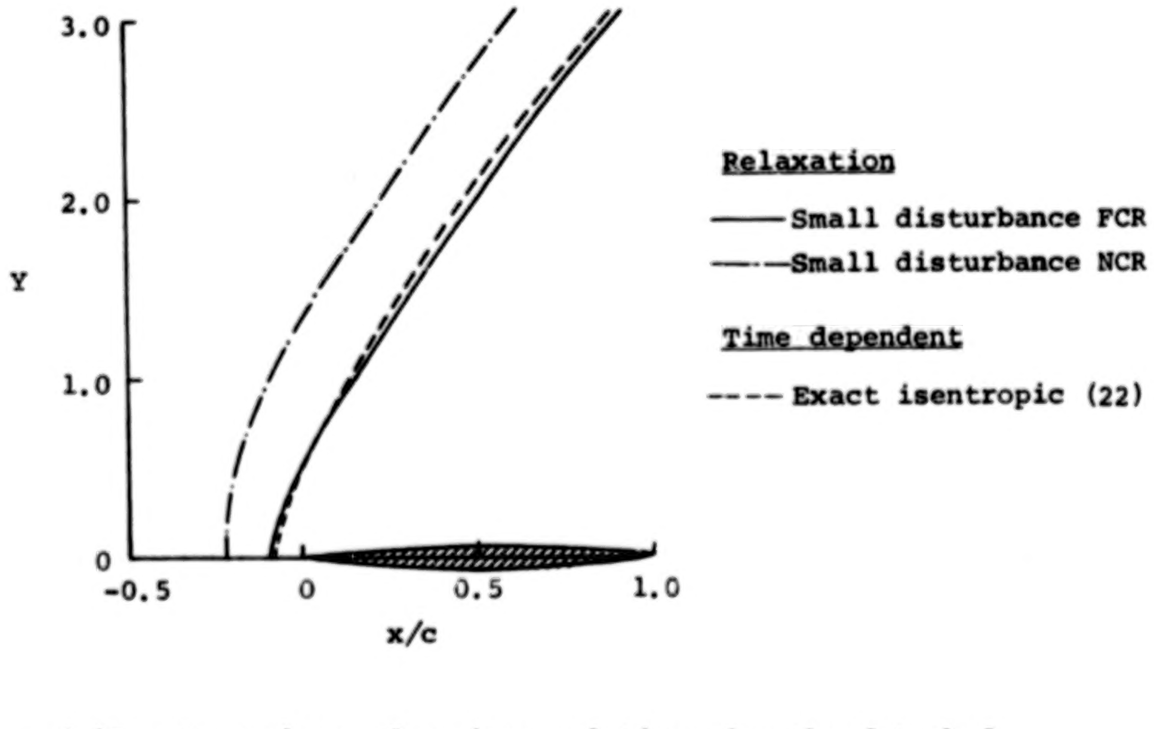


Figure 4-15.- Comparison of various solutions for the detached bow shock wave location for a parabolic arc airfoil with $\delta = 0.06$ and $M_\infty = 1.15$. (19)

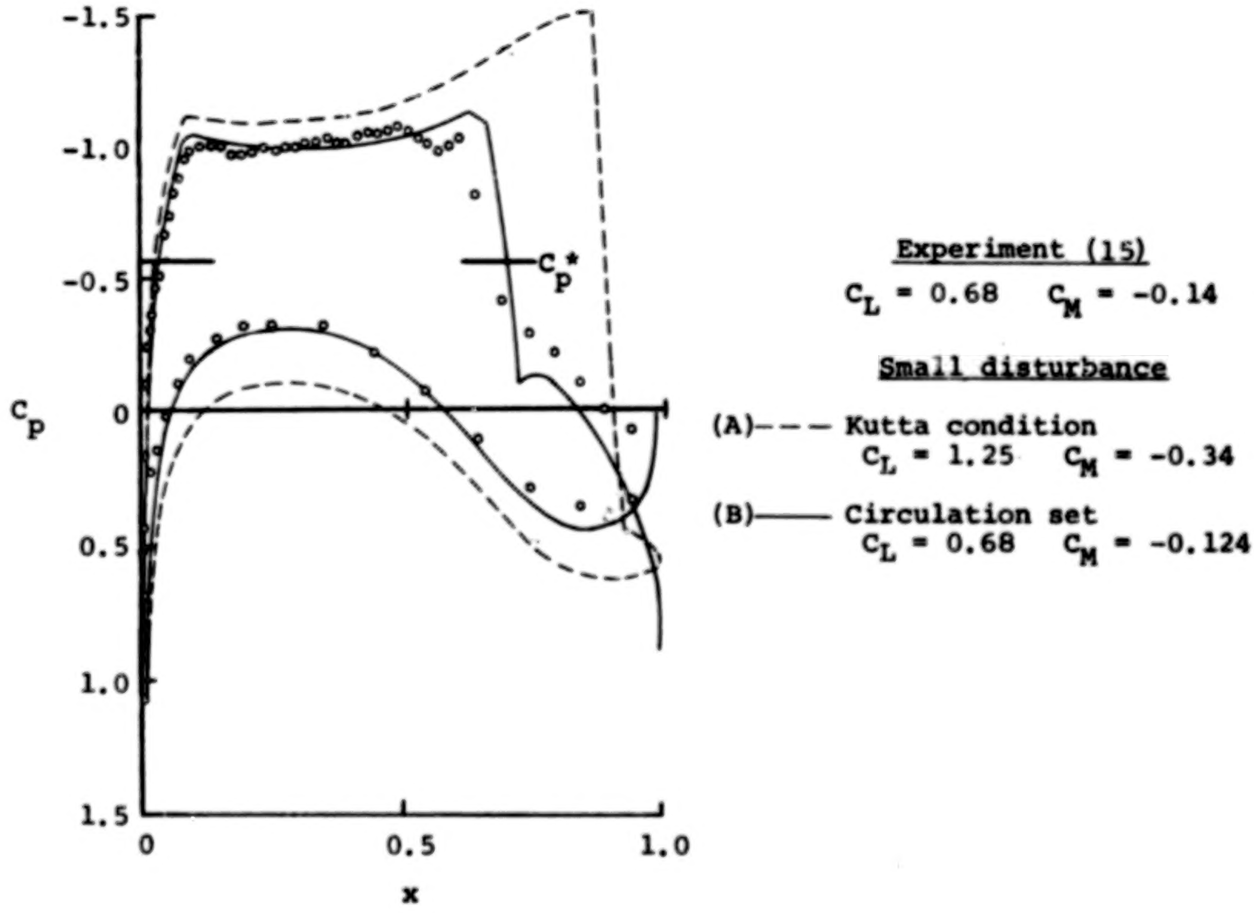


Figure 4-16.- Comparison of small disturbance FCR solutions with data for the surface pressure distribution on a Garabedian-Korn airfoil at $\alpha = 1.38^\circ$, $M_\infty = 0.768$. (19)

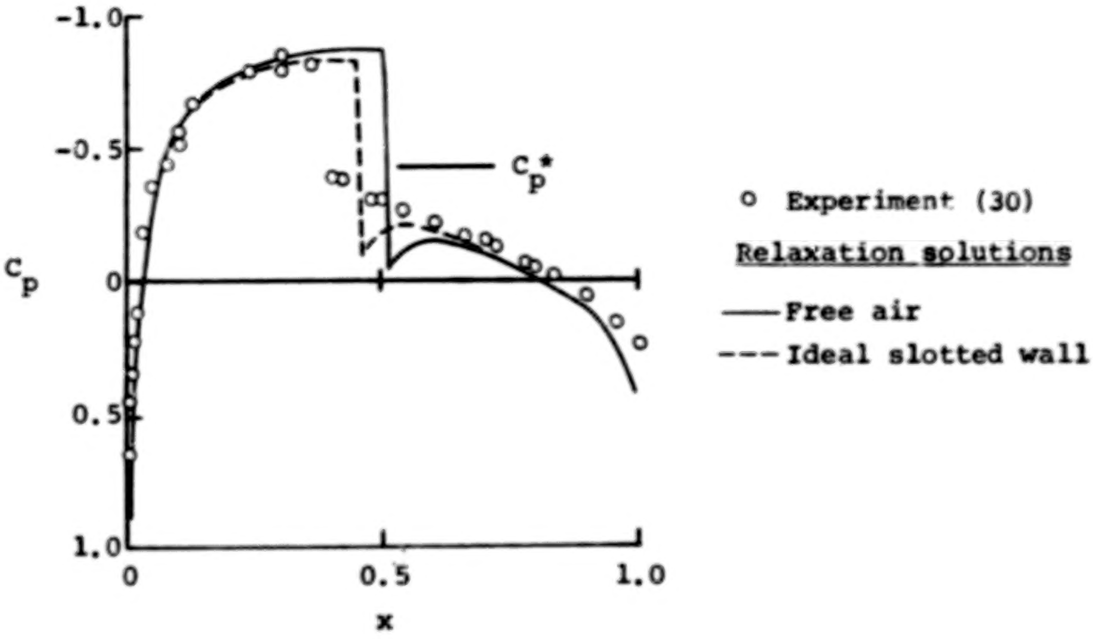


Figure 4-17.- Comparison of small disturbance FCR solutions with data for the surface pressure on a NACA 0012 at $\alpha = 0^\circ$ and $M = 0.8$. (19)

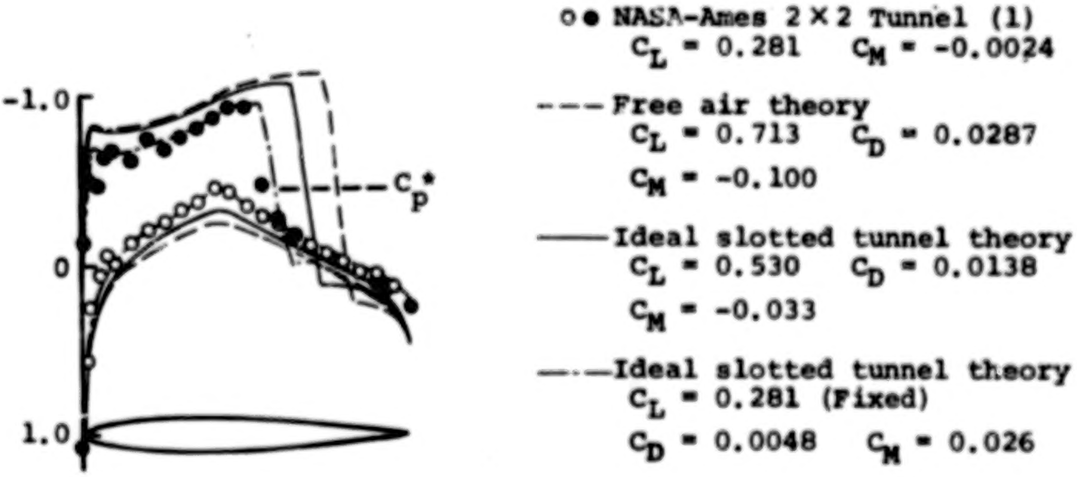


Figure 4-18.- Comparison of free-air and wind-tunnel small disturbance FCR solutions for the surface pressure distribution on a NACA 64A010 airfoil at $\alpha = 2.0^\circ$ and $M_\infty = 0.802$.

5. PROGRAMMER'S MANUAL

This section of the operational manual provides detailed information concerning the programming structure of the code, and is intended primarily to assist those users interested in modifying or extending the program. Included in this section are a dictionary of all subroutine variables that appear in common blocks in the program and a detailed description of the individual subroutines. These should provide all the information necessary to alter the code in a rapid and error-free fashion.

5.1 Dictionary of Subroutine Variables in Common

This section provides a dictionary of all subroutine variables which appear in common blocks in the program. The dictionary is arranged serially according to the named common block number rather than alphabetically in order to provide a more rapid access to the information.

VARIABLE	COMMON BLOCK	DESCRIPTION AND COMMENTS
	COMMON	
P(100,101)		array containing the small-disturbance velocity potential ϕ used in current calculation
X(100)		array containing the x-location of mesh points used in current calculation
Y(100)		array containing the y-location of mesh points used in current calculation
	COM 1	
IDOWN		downstream limit of I index for solution of difference equations; equal to IMAX - 1 for subsonic free streams, and IMAX otherwise
ILE		value of I index associated with the mesh point at the leading edge (x = 0.) or mesh point on the body closest to leading edge
IMAX		number of x-mesh points in current calculation
IMIN		value of I index designating the x-mesh point where the calculation is to start; user-option input, default value equal to 1

VARIABLE	COMMON BLOCK	DESCRIPTION AND COMMENTS
	COM 1 (cont'd)	
ITE		value of I index associated with the mesh point at the trailing edge ($x = 1.$) or mesh point on body closest to trailing edge
IUP		upstream limit of I index for solution of difference equations; equal to IMIN + 1 for subsonic free streams and IMIN + 2 otherwise
JBOT		lower limit of J index for solution of finite-difference equations; equal to JMIN + 1 for free-air subsonic calculations, free-jet calculations, and porous tunnel wall calculations where the scaled porosity is greater than 1.5; equal to JMIN otherwise
JLOW		value of J index associated with the row of mesh points immediately below the body
JMAX		number of y-mesh points in current calculation
JMIN		value of J index designating the y-mesh point where the calculation is to start; user-option input, default value equal to 1
JTOP		upper limit of J index for solution of finite-difference equations; equal to JMAX - 1 for free-air subsonic calculations, free-jet calculations, and porous tunnel wall calculations where the scaled porosity is greater than 1.5; equal to JMAX otherwise
JUP		value of J index associated with the row of mesh points immediately above the body
	COM 2	
AK		transonic similarity parameter; equal to $(1 - M_\infty^2)/M_\infty^{2m}\delta^{2/3}$ where m is equal to zero for Cole scaling, 2/3 for Spreiter scaling, and 1/2 for Krupp scaling
ALPHA		angle of attack; user-option input, default value equal to 0.12°
DUB		strength of doublet characterizing far-field behavior; set equal to airfoil volume for nonlifting free-air flows; equal to volume $+ (\gamma+1)/4 \iint \phi_x^2 \, dx \, dy$ otherwise
GAM1		equal to $\gamma + 1$
RTK		equal to $\sqrt{\lvert AK \rvert}$

VARIABLE	COMMON BLOCK	DESCRIPTION AND COMMENTS
	COM 3	
ABORT (logical)		program control for terminating calculation if solution diverges
ICUT		control for mesh cut and refinement; equal to 0, input mesh used to convergence; equal to 1, input mesh may be cut once; equal to 2, input mesh may be cut twice; user-option input, default value equal to 2
IREF		internal program control for mesh cutting
KSTEP		internal program control for index increment for mesh cutting
	COM 4	
XIN(100)		array of x-ordinates of user-supplied x-grid; set equal to XKRUPP if default x-grid chosen
YIN(100)		array of y-ordinates of user-supplied y-grid; set equal to YFREE or YTUN if default y-grid chosen
AMESH (logical)		control for use of analytical default mesh calculated by AYMESH; user-option input, default value is FALSE
	COM 5	
XDIFF(100)		array of difference coefficients for determining ϕ_x; equal to 1./(X(I) - X(I-1))
YDIFF(100)		array of difference coefficients for determining ϕ_y; equal to 1./(Y(J) - Y(J-1))
	COM 6	
CAMBER(100)		array of ordinates of airfoil camber-line distribution
FL(100)		array of y-ordinates of airfoil lower surface
FU(100)		array of y-ordinates of airfoil upper surface
FXL(100)		array of slopes (dF_ℓ/dx) of airfoil lower surface
FXU(100)		array of slopes (dF_u/dx) of airfoil upper surface
IFOIL		number of mesh points on airfoil; equal to ITE - ILE + 1

VARIABLE	COMMON BLOCK	DESCRIPTION AND COMMENTS
	COM 6 (con't)	
THICK(100)		array of ordinates of airfoil thickness distribution
VOL		volume of airfoil/unit span
XFOIL(100)		array of x-ordinates of mesh points on airfoil
	COM 7	
CJLOW		difference coefficient for extrapolation formulas to obtain airfoil lower surface properties; equal to -Y(JLOW-1)/(Y(JLOW) - Y(JLOW-1))
CJLOW1		difference coefficient for extrapolation formulas to obtain airfoil lower surface properties; equal to -Y(JLOW)/(Y(JLOW) - Y(JLOW-1))
CJUP		difference coefficient for extrapolation formulas to obtain airfoil upper surface properties; equal to Y(JUP+1)/(Y(JUP+1) - Y(JUP))
CJUP1		difference coefficient for extrapolation formulas to obtain airfoil upper surface properties; equal to Y(JUP)/(Y(JUP+1) - Y(JUP))
	COM 8	
CVERGE		convergence criterion for maximum iteration error of ϕ; user-option input, default value equal to 0.00005
DVERGE		divergence criterion for maximum iteration error ϕ; user-option input, default value equal to 10.
EPS		coefficient of pseudo-time term $\Delta t\phi_{xt}/\Delta x$ in differential equation; value between 0. and 1.; default value equal to 0.
IPRTER		control for print frequency of y-line in mesh where error is largest; for example, IPRTER = 10 implies line will be printed every 10th iteration; user-option input, default value equal to 10
MAXIT		control for maximum number of iteration cycles allowed on each mesh; user-option input, default value equal to 500

VARIABLE	COMMON BLOCK	DESCRIPTION AND COMMENTS
	COM 8 (cont'd)	
WE(3)		relaxation factor at elliptic points on the coarse, medium, and fine mesh, respectively; user-option input, default values in order are 1.8, 1.9, and 1.95; if user-specified, all three values must be given
	COM 9	
BCFOIL (integer)		control for airfoil geometry options; equal to 1 for symmetric NACA four-digit (00XX) series; equal to 2 for parabolic arc; equal to 3 for user-specified ordinates, program calculates ordinates and slopes at mesh points by cubic spline interpolation; equal to 4 for complete user-supplied description; user-option input, default value is 3 and program uses Korn airfoil ordinates
NL		number of ordinate points used to describe airfoil lower surface; default value (for Korn airfoil) is 75
NU		number of ordinate points used to describe airfoil upper surface; default value (for Korn airfoil) is 100
RIGF		Reigles rule nose correction factor for modifying airfoil slopes; user-option input, default value is 0.0
XL(100)		array of x-ordinates for airfoil lower surface; user-supplied if BCFOIL = 3, unless default (Korn) airfoil is used
XU(100)		array of x-ordinates for airfoil upper surface; user-supplied if BCFOIL = 3, unless default (Korn) airfoil is used
YL(100)		array of y-ordinates for airfoil lower surface; user-supplied if BCFOIL = 3, unless default (Korn) airfoil is used
YU(100)		array of y-ordinates for airfoil upper surface; user-supplied if BCFOIL = 3, unless default (Korn) airfoil is used
	COM 10	
GAM		ratio of specific heats; user-option input, default value equal to 1.4
JMXF		number of y-mesh points for basic Krupp free-air grid; equal to 57
JMXT		number of y-mesh points for basic wind-tunnel grid; equal to 49

VARIABLE	COMMON BLOCK	DESCRIPTION AND COMMENTS
	COM 10 (cont'd)	
XGRDIN		control for input of x-grid; equal to TRUE if x-grid is user-specified; equal to FALSE if default x-grid (XKRUPP) is used; default value equal to FALSE
XKRUPP(100)		array of x-ordinates for basic Krupp free-air grid
YFREE(100)		array of y-ordinates for basic Krupp free-air grid
YGRDIN		control for input of y-grids; equal to TRUE if y-grid is user-speicified; equal to FALSE if either default y-grid (YFREE or YTUN) is used; default value equal to FALSE
YTUN(100)		array of y-ordinates for basic wind-tunnel grid
	COM 11	
ALPHAO		value of angle of attack (ALPHA) from previous calculation
CLOLD		value of lift coefficient (CL) from previous calculation
DELTAO		value of thickness ratio (DELTA) from previous calculation
DUBO		value of far-field doublet strength (DUB) from previous calculation
EMACHO		value of Mach number (EMACH) from previous calculation
IMAXI		number of x-mesh points; less than or equal to 100; user-option input, default value is 77 for basic Krupp grid
IMAXO		number of x-mesh points (IMAX) used in previous calculation
IMINO		value of I index (IMIN) where previous calculation was initiated
JMAXI		number of y-mesh points; less than or equal to 100; user-option input, default value is 57 (JMXF) for basic Krupp free-air grid or 49 (JMXT) for basic wind-tunnel grid

VARIABLE	COMMON BLOCK	DESCRIPTION AND COMMENTS
	COM 11 (cont'd)	
JMAXO		number of y-mesh points (JMAX) used in previous calculation
JMINO		value of J index (JMIN) where calculation was initiated
PSAVE (logical)		control for saving restart block of values on unit 3; equal to TRUE for final results of current calculation to be saved; user-option input, default value is FALSE
PSTART (integer)		control for initializing velocity potential ϕ array; equal to 1, ϕ set to zero; equal to 2, ϕ read from tape unit 7; equal to 3, ϕ used from previous case; user-option input, default value equal to 1
TITLE(8)		user-input alphanumeric array containing run title information
TITLEO(8)		alphanumeric array containing run title information from previous run
VOLO		value of airfoil volume (per unit span) from previous calculation
XOLD(100)		array of x-mesh ordinates from previous calculation
YOLD(100)		array of y-mesh ordinates from previous calculation
	COM 12	
F		wind-tunnel slot parameter; user-option input, default value equal to 0.
H		wind-tunnel half height to chord ratio; user-option input, default value equal to 0.
HALFPI		equal to $\pi/2$
PI		equal to π
RTKPOR		equal to $\sqrt{\lvert AK \rvert}/POR$
TWOPI		equal to 2π

VARIABLE	COMMON BLOCK	DESCRIPTION AND COMMENTS
	COM 13	
CDFACT		transonic scale factor for drag coefficient; equal to $\delta^{4/3}M_\infty^{-n}$ where n equals 0 for Cole scaling, 2/3 for Spreiter scaling, and 3/4 for Krupp scaling
CLFACT		transonic scale factor for lift coefficient; equal to $\delta^{2/3}M_\infty^{-n}$ (see CDFACT)
CMFACT		transonic scale factor for moment coefficient; equal to $\delta^{2/3}M_\infty^{-n}$ (see CDFACT)
CPFACT		transonic scale factor for pressure coefficient; equal to $\delta^{2/3}M_\infty^{-n}$ (see CDFACT)
CPSTAR		critical pressure coefficient
	COM 14	
CLSET		specified lift coefficient; used if Kutta condition is FALSE; user-option input, default value equal to 0.
FCR (logical)		control for fully conservative differencing; equal to TRUE for fully conservative form; otherwise, difference equations not conservative at shock waves; user-option input, default value equal to TRUE
KUTTA (logical)		control for Kutta condition specification; equal to TRUE for Kutta condition enforcement; equal to FALSE for lift-coefficient specification; user-option input, default value equal to TRUE
WCIRC		relaxation factor for circulation; user-option input, default value equal to 1.
	COM 15	
B		constant in subsonic far-field doublet solution for various wind-tunnel walls; equal to 0.5 for solid wall; equal to 0. for free-jet, slotted, or perforated/porous walls
BETA0		constant in subsonic far-field vortex solution for various wind-tunnel walls; calculated by VROOTS for ideal slotted tunnel wall; equal to $\pi/2$ for solid wall; equal to 0. for free jet; equal to $\tan^{-1}$(RTK/POR) for ideal perforated/porous wall

VARIABLE	COMMON BLOCK	DESCRIPTION AND COMMENTS
	COM 15 (cont'd)	
BETA1		see BETA0
BETA2		see BETA0
PSI0		constant in subsonic far-field vortex solution for various wind-tunnel walls; calculated by VROOTS for ideal slotted tunnel wall; equal to 1. for solid wall, free-jet, and ideal perforated/porous wall
PSI1		see PSI0
PSI2		see PSI0
	COM 16	
ALPHA0		constant in subsonic far-field doublet solution for various wind-tunnel walls; calculated by DROOTS for ideal slotted tunnel wall; equal to π for solid wall; equal to $\pi/2$ for free-jet; equal to $\pi/2 - \tan^{-1}$(RTK/POR) for ideal perforated/porous wall
ALPHA1		see ALPHA0
ALPHA2		see ALPHA0
JET (real)		constant in subsonic far-field vortex solution for various wind-tunnel walls; equal to 0. for solid and ideal perforated/porous wall; equal to 0.5 for free-jet and ideal slotted walls
OMEGA0		constant in subsonic far-field doublet solution for various wind-tunnel walls; calculated by DROOTS for ideal slotted tunnel wall; equal to 1. for solid wall, free-jet, and ideal perforated/porous walls
OMEGA1		see OMEGA0
OMEGA2		see OMEGA0
XSING		x-location of vortex and doublet used to determine far-field behavior on outer boundaries; equal to 0.5
	COM 17	
CYYBLC		special ϕ_{yy} difference coefficient at airfoil lower surface; equal to -CYYBLU/(Y(JLOW) - Y(JLOW-1))

VARIABLE	COMMON BLOCK	DESCRIPTION AND COMMENTS
	COM 17 (cont'd)	
CYYBLD		special ϕ_{yy} difference coefficient at airfoil lower surface; equal to -CYYBLC
CYYBLU		special ϕ_{yy} difference coefficient at airfoil lower surface; equal to -2./(Y(JLOW) + Y(JLOW-1))
CYYBUC		special ϕ_{yy} difference coefficient at airfoil upper surface; equal to CYYBUD/(Y(JUP+1) - Y(JUP))
CYYBUD		special ϕ_{yy} difference coefficient at airfoil upper surface; equal to -2./(Y(JUP+1) + Y(JUP))
CYYBUU		special ϕ_{yy} difference coefficient at airfoil upper surface; equal to -CYYBUC
FXLBC(100)		array containing body-slope boundary condition on lower surface multiplied by mesh spacing constant CYYBLU
FXUBC(100)		array containing body-slope boundary condition on upper surface multiplied by mesh spacing constant CYYBUD
ITEMP1		dummy index
ITEMP2		dummy index
	COM 18	
DCIRC		iteration error in circulation; equal to $\Delta\phi_{te}^{n+1} - \Delta\phi_{te}^{n}$
ERROR		absolute value of the maximum iteration error in ϕ; that is, $\|\phi_{j,i}^{n} - \phi_{j,i}^{n-1}\|_{max}$
I1		dummy I index used to alternate storage in arrays POLD (J,I) and EMU (J,I)
I2		dummy I index used to alternate storage in arrays POLD (J,I) and EMU (J,I)
IERROR		value of I index where maximum iteration error in ϕ occurs
JERROR		value of J index where maximum iteration error in ϕ occurs
OUTERR (logical)		internal program control for printout of iteration count, CL, CM, CD, IERROR, JERROR, ERROR, and ERCIRC
POLD(100,2)		array used to store current and previous column of $\phi_{j,i}$

VARIABLE	COMMON BLOCK	DESCRIPTION AND COMMENTS
	COM 18 (cont'd)	
VC(100)		array used to store current column of $2\cdot(1-M_I^2)/(X_{i+1}-X_{i+1})$; equal to C1(I) - (CXL(I)·POLD(J,I2) + CXC(I)·P(J,I) + CXR(I)·P(J,I+1))
	COM 19	
WI		reciprocal of elliptic relaxation factor for current mesh
DIAG(100)		array of diagonal elements of tridiagonal matrix equation for column solution of ϕ
RHS(100)		array for the right-hand side column vector of tridiagonal matrix equation for column solution of ϕ
SUB(100)		array of sub-diagonal elements of tridiagonal matrix equation for column solution of ϕ
SUP(100)		array of super-diagonal elements of tridiagonal matrix equation for column solution of ϕ
	COM 20	
XMID(100)		array of x-mesh points obtained by deleting every other point in input array XIN starting with I = IMIN + 1
YMID(100)		array of y-mesh points obtained by deleting every other point in input array YIN starting with J = JMIN + 1
	COM 22	
CXC(100)		centered ϕ_x difference coefficient devided by X(I+1) - X(I-1) for IMIN + 1 $\leq$ I $\leq$ IMAX - 1; equal to -CXL(I) - CXR(I)
CXL(100)		left-biased ϕ_x difference coefficient divided by X(I+1) - X(I-1) for IMIN + 1 $\leq$ I $\leq$ IMAX - 1; equal to -0.5*(γ+1)/((X(I) - X(I-1))*(X(I+1) - X(I-1)))
CXR(100)		right-biased ϕ_x difference coefficient divided by X(I+1) - X(I-1) for IMIN + 1 $\leq$ I $\leq$ IMAX - 1; equal to 0.5*(γ+1)/((X(I+1) - X(I))*(X(I+1) - X(I-1)))
CXXC(100)		centered ϕ_{xx} difference coefficient multiplied by X(I+1) - X(I-1) for IMIN + 1 $\leq$ I $\leq$ IMAX - 1; equal to -CXXL(I) - CXXR(I)

VARIABLE	COMMON BLOCK	DESCRIPTION AND COMMENTS
	COM 22 (cont'd)	
CXXL(100)		left-biased ϕ_{xx} difference coefficient multiplied by X(I+1) - X(I-1) for IMIN + 1 $\leq$ I $\leq$ IMAX - 1; equal to 2./(X(I) - X(I-1))
CXXR(100)		right-biased ϕ_{xx} difference coefficient multiplied by X(I+1) - X(I-1) for IMIN + 1 $\leq$ I $\leq$ IMAX - 1; equal to 2./(X(I+1) - X(I))
C1(100)		equal to AK/(X(I+1) - X(I-1))
	COM 23	
CYYC(100)		centered ϕ_{yy} difference coefficient; equal to -CYYD(J) - CYYU(J) for JMIN + 1 $\leq$ J $\leq$ JMAX - 1
CYYD(100)		down-biased ϕ_{yy} difference coefficient; equal to 2./((Y(J) - Y(J-1))*(Y(J+1) - Y(J-2))) for JMIN + 1 $\leq$ J $\leq$ JMAX - 1
CYYU(100)		up-biased ϕ_{yy} difference coefficient; equal to 2./((Y(J+1) - Y(J))*(Y(J+2) - Y(J-2))) for JMIN + 1 $\leq$ J $\leq$ JMAX - 1
IVAL		index identifying the current column location; equal to I
	COM 24	
DBOT(100)		array of subsonic, far-field, unit-strength doublet values on bottom boundary
DDOWN(100)		array of subsonic, far-field, unit-strength doublet values on downstream boundary
DTOP(100)		array of subsonic, far-field, unit-strength doublet values on top boundary
DUP(100)		array of subsonic, far-field, unit-strength doublet values on upstream boundary
VBOT(100)		array of subsonic, far-field, unit-strength vortex values on bottom boundary
VDOWN(100)		array of subsonic, far-field, unit-strength vortex values on downstream boundary
VTOP(100)		array of subsonic, far-field, unit-strength vortex values on top boundary
VUP(100)		array of subsonic, far-field, unit-strength vortex values on upper boundary

VARIABLE	COMMON BLOCK	DESCRIPTION AND COMMENTS
	COM 25	
CPL(100)		array of pressure coefficient values on $y = 0^-$ lines from $\mathrm{IMIN} \leq \mathrm{I} \leq \mathrm{IMAX}$
CPU(100)		array of pressure coefficient values on $y = 0^+$ line from $\mathrm{IMIN} \leq \mathrm{I} \leq \mathrm{IMAX}$
	COM 26	
PJUMP(100)		array of values for the jump in potential across the wake from $\mathrm{ITE} \leq \mathrm{I} \leq \mathrm{IMAX}$; equal to CIRCTE + (X(I) - 1.)*(CIRCFF-CIRCTE)/X(IMAX1 - 1.)
	COM 27	
CL		lift coefficient
DELRT2		equal to $(\delta)^{2/3}$
DELTA		airfoil thickness ratio δ; user-option input, default value equal to 0.115
EMACH		free-stream Mach number M_∞; user-option input, default value equal to 0.75
EMROOT		equal to $(M_\infty)^{2/3}$
PHYS (logical)		control for transonic scaling used for input/output; equal to TRUE for physically scaled values and FALSE for transonically scaled values; user-option input, default value equal to TRUE
PRTFLO (integer)		control for print option of final flow field; equal to 1 for no flow field printout, equal to 2 for entire flow field printout, equal to 3 for printout of 3J lines around maximum error; user-option input, default value equal to 1
SIMDEF (integer)		control for transonic similarity scaling definition; equal to 1 for Cole scaling, equal to 2 for Spreiter scaling, equal to 3 for Krupp scaling, equal to 4 for user-supplied scaling; user-option input, default value equal to 3
SONVEL		perturbation velocity at sonic point; equal to AK/GAM1
VFACT		transonic scale factor for flow angle deflection; equal to $\delta\cdot(180^\circ/\pi)$

VARIABLE	COMMON BLOCK	DESCRIPTION AND COMMENTS
	COM 27 (cont'd)	
YFACT		transonic scale factor for lateral coordinate; equal to $\delta^{-1/3} M_\infty^{-m}$ where m equals 0 for Cole scaling, 2/3 for Spreiter scaling, and 1/2 for Krupp scaling
	COM 28	
BCTYPE (integer)		input control for type of flow to be computed; equal to 1 for free air, 2 for solid wall tunnels, 3 for free jet, 4 for ideal slotted-wall tunnel, and 5 for ideal perforated/porous wall tunnel; user-option input, default value equal to 1
CIRCFF		circulation in far field
CIRCTE		jump in potential at the trailing edge, $\Delta\phi_{te}$
FHINV		equal to 1./(F*H)
POR		wall porosity factor; user-option input, default value equal to 0.
	COM 30	
		common block COM 30 is a scratch common used to shorten storage requirements; all variables appearing in COM 30 are used internally only within each subroutine and not passed for use to other subroutines
	COM 32	
BIGRL		maximum residual of finite difference equation; equal to $(R_{j,i})_{max}$
IRL		value of I index where maximum residual occurs
JRL		value of J index where maximum residual occurs
	COM 33	
THETA(100,100)		array containing values of the angle used to update the potential for subsonic free-air flows by spreading the circulation rapidly through the flow field; calculated by ANGLE; equal to $1/2\pi \tan^{-1}(\sqrt{K}\,\tilde{y}/(X_i - \mathrm{XSING}))$
	SPLN	
A(200)		coefficient array for cubic-fit, continuous derivative interpolation formula for fitting tabulated airfoil ordinates

VARIABLE	COMMON BLOCK	DESCRIPTION AND COMMENTS
	SPLN (cont'd)	
B(200)		coefficient array for cubic-fit, continuous derivation interpolation formula for fitting tabulated airfoil ordinates
DY1		slope of airfoil upper or lower surface at leading edge
DY2		slope of airfoil upper or lower surface at trailing edge
DYP		slope of airfoil at XP calculated by cubic spline interpolation SPLN1
K1		index for internal control of SPLN1 subroutine; equal to 1 implies first derivative specified, equal to 2 implies second derivative specified
K2		index for internal control of SPLN1 subroutine; equal to 1 implies first derivative specified, equal to 2 implies second derivative specified
XP		x-location of current mesh point on airfoil XIN(I), ILE $\leq$ I $\leq$ ITE
YP		y-ordinate of airfoil at XP calculated by cubic spline interplation

5.2 Description of Individual Subroutines

This section provides a description of the individual subroutines in the program, including the main program. The descriptions are ordered alphabetically according to subroutine name.

5.2.1 Subroutine ANGLE.- Subroutine ANGLE computes the quantity THETA(J,I) at the input mesh points (I,J) according to

$$\mathrm{THETA(J,I)} = \frac{1}{2\pi} \tan^{-1}\left(\frac{\sqrt{K}\,\tilde{y}_j}{x_i - \mathrm{XSING}}\right)$$

where XSING = 0.5. For $R = \sqrt{\tilde{y}_j^2 + (x_i - \mathrm{XSING})^2} \leq 1$, the quantity is reduced by THETA(J,I) = R*THETA(J,I). These values are used in subroutine SOLVE to enhance the convergence characteristics of subsonic free-air flows by transmitting the effect of updated circulation throughout the flow field. This is accomplished by updating the potential at the end of each iteration sweep in x by

$$\phi_{j,i}^{n+1} = \phi_{j,i}^{n+1} + (\Gamma^{n+1} - \Gamma^{n})\cdot \mathrm{THETA(J,I)}$$

5.2.2 Subroutine ARF(X).- Subroutine ARF(X) evaluates the error function

$$\mathrm{ARF(X)} = \frac{2}{\sqrt{\pi}}\int_0^X e^{-t^2}dt$$

by using the rational fraction approximation given in the Handbook of Mathematical Functions, AMS 55, edited by M. Abramowitz and L. A. Stegun, Fifth Printing, August 1966, p. 299, Equation 7.1.26.

5.2.3 Subroutine AYMESH.- Subroutine AYMESH computes an analytical $(\mathbf{x},\tilde{y})$ mesh based in part on the formulation in reference 11. That mesh formulation was designed for $M_\infty < 1$ free-air flows; however, it can be used for $M_\infty > 1$ flows as well as for wind tunnel environments. A coordinate tranformation in $\mathbf{x}$ is made according to the following formula:

$$X(\mathbf{x}) = (1 - a_0)\left\{a_1 x e^{-(a_2 x)^2} + \left[1 - e^{-(a_7 x)^2}\right]\frac{2}{\sqrt{\pi}}\int_0^{a_4 x} e^{-t^2}dt\right\}$$

$$+ \frac{2a_6}{\pi}\tan^{-1}\left[a_5(x + a_0)\right]$$

An initial $\mathbf{x}^{(1)}$ mesh is chosen by setting

$$\mathbf{x}_I^{(1)} = \tan\left[\frac{\pi}{2}\left(\frac{I - 201}{200}\right)\right]_{I=201}^{401}$$

$$\mathbf{x}_I^{(1)}\Big]_{I=1}^{200} = -\mathbf{x}_{402-I}^{(1)}$$

Corresponding $X_I^{(1)}(\mathbf{x}_I^{(1)})$ are then computed from the above stretching with the following coefficient values: $a_0 = 0.225$, $a_1 = 1.4$, $a_2 = 1.6$, $a_4 = 0.75$, $a_5 = 30.0$, $a_6 = 0.603$, and $a_7 = 2.0$. Then, for a fixed ΔX increment equal to $\Delta X = 2/(\mathrm{IMAXI}-1)$, the corresponding $\mathbf{x}_i$ points are found by using a quadratic interpolation scheme with the $X^{(1)}$, $\mathbf{x}^{(1)}$ arrays. The final $\mathbf{x}_i$ mesh is found by using $\mathbf{x}_i = \mathbf{x}_i + a_3$ where $a_3 = 0.6188$.

The $\tilde{y}$ array definition depends on whether the calculation is for free air or a tunnel environment. For free air

$$\tilde{y}_j = 5.3\,\frac{\tan\left[\frac{\pi}{2}\cdot\frac{j-\frac{JMAXI}{2}}{\left(\frac{JMAXI}{2}+1\right)}\right]}{\tan\left[\frac{\pi}{2}\cdot\frac{\frac{JMAXI}{2}}{\frac{JMAXI}{2}+1}\right]}\,, \quad \frac{JMAX}{2}+1 \le j \le JMAXI$$

$$\tilde{y}_j = -\tilde{y}_{JMAXI+1-j}\,, \quad 1 \le j \le \frac{JMAXI}{2}$$

while for a tunnel environment

$$\tilde{y}_j = \tan^2\left[\frac{\pi}{2}\cdot\frac{j-\frac{JMAXI}{2}}{JMAXI}\right], \quad \frac{JMAX}{2}+1 \le j \le JMAXI$$

$$\tilde{y}_j = -\tilde{y}_{JMAXI+1-j}\,, \quad 1 \le j \le \frac{JMAXI}{2}$$

The default values for IMAXI and JMAXI used with this analytical mesh are IMAXI = 81, JMAXI = 64. These values produce a mesh sufficiently fine for almost all aerodynamic applications and permit the mesh halving option to be used twice; that is, fine to medium to coarse.

5.2.4 Subroutine BCEND (I,JEND,L).- Subroutine BCEND modifies the diagonal DIAG(J) and right-hand side RHS(J) elements of the tridiagonal matrix in the tridiagonal matrix equation for the residual of the velocity potential along a column to include the boundary conditions along the bottom (J = JBOT) and top (J = JTOP) rows. The routine first sets the index I (I = IVAL) to correspond to the current column being solved by SYOR and then branches to the appropriate modification of the type of flow environment being considered according to the following schedule:

For the top boundary,

BCTYPE	JTOP	DIAG MODIFICATION	RHS MODIFICATION
1(K) > 0)	JMAXI-1	----	----
1(K) < 0)	JMAX	$\frac{-\sqrt{-K}\cdot \mathrm{CYYU(JTOP)}}{x_i - x_{i-1}}$	$\frac{\sqrt{-K}\cdot \mathrm{CYYU(JTOP)}\cdot(\phi_{JMAX,i} - \phi_{JMAX,i-1})}{x_i - x_{i-1}}$ $+ \mathrm{CYYU(JTOP)}\cdot\phi_{JTOP+1,i}$
2	JMAX	----	----
3(K > 0)	JMAX-1	----	$\mathrm{CYYU(JTOP)}\left(\frac{\Gamma_{ff}}{4} + \phi_{JTOP+1,i}\right)$
3(K < 0)	JMAX-1	----	$\mathrm{CYYU(JTOP)}\cdot\phi_{JTOP+1,i}$
4(K > 0)	JMAX	$\frac{-\ \mathrm{CYYU(JTOP)}}{\mathrm{FH}}$	$\mathrm{CYYU(JTOP)}\left[\frac{1}{\mathrm{FH}}\left(\frac{\Gamma_{ff}}{4} + \phi_{JTOP,i}\right) + \phi_{JTOP,1,i}\right]$
4(K < 0)	JMAX	$\frac{-\ \mathrm{CYYU(JTOP)}}{\mathrm{FH}}$	$\mathrm{CYYU(JTOP)}\left[\frac{1}{\mathrm{FH}}\phi_{JTOP,i} + \phi_{JTOP+1,i}\right]$
5(POR > 1.5)	JMAX-1	----	$\mathrm{CYYU(JTOP)}\left(\phi^{(n-1)}_{JMAX,i} - \phi^{(n)}_{JTOP+1,i}\right)$ ($\phi_{JMAX,i}$ from eqs. (A), (B))
5(POR < 1.5)	JMAX	$-\ \frac{\mathrm{POR}\cdot\mathrm{CYYU(JTOP)}}{x_i - x_{i-1}}$	$\frac{\mathrm{POR}\cdot\mathrm{CYYU(JTOP)}(\phi_{JMAX,i} - \phi_{JMAX,i-1})}{x_i - x_{i-1}}$ $+ \mathrm{CYYU(JTOP)}\cdot\phi_{JTOP+1,i}$

and for the bottom boundary

BCTYPE	JBOT	DIAG MODIFICATION	RHS MODIFICATION
1(K > 0)	JMIN+1	----	----
1(K < 0)	JMIN	$\frac{-\sqrt{-K}\cdot CYYD(JBOT)}{x_i - x_{i-1}}$	$\frac{\sqrt{-K}\cdot CYYD(JBOT)\cdot\phi_{JMIN,i} - \phi_{JMIN-1,i}}{x_i - x_{i-1}}$ $+ CYYD(JBOT)\cdot\phi_{JBOT-1,i}$
2	JMIN	----	----
3(K > 0)	JMIN+1	----	$CYYD(JBOT)\left(\frac{3\Gamma_{ff}}{4} + \phi_{JBOT-1,i}\right)$
3(K < 0)	JMIN+1	----	$CYYD(JBOT)\cdot\phi_{JBOT-1,i}$
4(K > 0)	JMIN	$\frac{-CYYD(JBOT)}{FH}$	$CYYD(JBOT)\left[\frac{1}{FH}\left(\frac{3\Gamma_{ff}}{4} + \phi_{JBOT,i}\right) + \phi_{JBOT-1,i}\right]$
4(K < 0)	JMIN	$\frac{-CYYD(JBOT)}{FH}$	$CYYD(JBOT)\left[\frac{1}{FH}\phi_{JBOT,i} + \phi_{JBOT-1,i}\right]$
5(POR > 1.5)	JMIN+1	----	$CYYD(JBOT)\left(\phi^{(n-1)}_{JMIN,i} - \phi^{(n)}_{JBOT-1,i}\right)$ ($\phi_{JMIN,i}$ from eqs. (A), (B))
5(POR < 1.5)	JMIN	$\frac{-POR\cdot CYYD(JBOT)}{x_i - x_{i-1}}$	$\frac{POR\cdot CYYD(JBOT)(\phi_{JMIN,i} - \phi_{JMIN,i-1})}{x_i - x_{i-1}}$ $+ CYYD(JBOT)\cdot\phi_{JBOT-1,i}$

where for BCTYPE = 5, superscript n refers to the iteration count,

$$\phi^{(n)}_{j,i} = \phi^{(n)}_{j,i-1} \mp \frac{x_i - x_{i-1}}{POR} (\phi_y^{(n)})_{j,i-1/2} \qquad \text{(A)}$$

and

$$(\phi_y^{(n)})_{\substack{JMAX\\JMIN},i-1/2} = \frac{1}{2}\left\{ \frac{\phi^{(n)}_{\substack{JMAX\\JMIN},i} - \phi^{(n)}_{\substack{JMAX-1\\JMIN+1},i}}{Y_{\substack{JMAX\\JMIN}} - Y_{\substack{JMAX-1\\JMIN+1}}} + \frac{\phi^{(n)}_{\substack{JMAX\\JMIN},i-1} - \phi^{(n)}_{\substack{JMAX-1\\JMIN+1},i-1}}{Y_{\substack{JMAX\\JMIN}} - Y_{\substack{JMAX-1\\JMIN+1}}} \right\} \qquad \text{(B)}$$

where the (-) sign in equation (A) corresponds to the top (J = JTOP) boundary. This modification applies only for i = IUP, at which point all of the $\phi^{(n)}_{JMAX,i}$ and $\phi^{(n)}_{JMIN,i}$ values are set for the current iteration x sweep. No modification is necessary for i > IUP.

For BCTYPE = 6, the difference equations have not yet been worked out. Consequently, the user must insert the appropriate information required; otherwise, an error message is printed and an abnormal stop occurs.

5.2.5 Subroutine BLOCK DATA.- Subroutine BLOCK DATA is a storage subroutine which contains all of the default values for the various parameters and arrays required as input to the program.

5.2.6 Subroutine BODY.- Subroutine BODY determines the body geometry information required for the boundary conditions and output. Four user-option choices of body description are programmed and available according to the input control BCFOIL as follows:

BCFOIL	AIRFOIL
1	NACA 00XX series
2	Parabolic-arc
3	User-supplied ordinates (default = Korn)
4	User specification

The subroutine computes the body ordinates and slopes at the input x-mesh locations and normalizes them by the thickness ratio δ. For BCFOIL = 3, a cubic spline interpolation is used for both slopes and ordinates, Next, the body volume per unit span, camber and thickness distributions are computed, then the airfoil upper and lower airfoil slopes are recomputed using a Reigel's rule nose correction RIGF according to

$$\mathrm{FXU(I)} = \frac{\mathrm{FXU(I)}}{\sqrt{1 + \mathrm{RIGF}(\lambda\cdot\mathrm{FXU(I)})^2}}$$

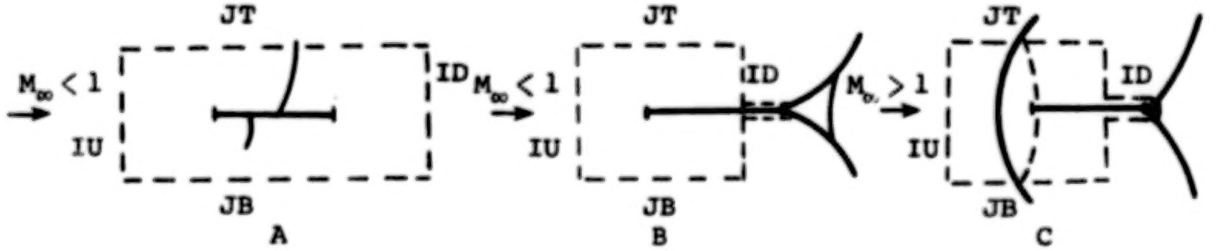
$$\mathrm{FXL(I)} = \frac{\mathrm{FXL(I)}}{\sqrt{1 + \mathrm{RIGF}(\lambda\cdot\mathrm{FXL(I)})^2}}$$

where $0 \leq \mathrm{RIGF} \leq 1$ and is user-specified. Finally, subroutine PRBODY is called to output the calculated information.

5.2.7 Subroutine CDCOLE.- Subroutine CDCOLE computes the airfoil inviscid drag by the momentum integral method; that is, by integrating around a contour enclosing the body and along all shocks inside the contour according to the following formulation:

$$\frac{C_D}{2\delta^{5/3}M_\infty^{-n}} = \oint_C \left[\left(K\frac{u^2}{2} - \frac{v^2}{2} - \frac{\gamma+1}{3}u^3\right)d\tilde{y} - (uv)\,dx\right] - \frac{\gamma+1}{12}\int_{S\supset C} [u]^3 d\tilde{y}$$

where $\tilde{y}$ is the transonically scaled y coordinate, n depends on the transonic scaling used (see Table 4-I), and $[u]$ is the jump in ϕ_x across the shock waves S contained within the contour C. Sample contours incorporated in the program are shown below:



where for the subsonic contours A and B, the upstream boundary is set at IU = (IMIN + ILE)/2 and the top and bottom boundaries at JT = JMAX - 1 and JB = JMIN + 1, respectively. For situations where the velocity at the upper surface of the trailing edge is less than sonic, as in contour A, the downstream boundary is set at ID = (IMAX + ITE)/2. Otherwise, ID is the I index at the x-mesh point closest to the three-quarter chord point on the airfoil, $X(I) \leq 0.75$, as shown in contours B and C. For supersonic free streams, as in contour C, the upper and lower boundaries are set to encompass the subsonic region ahead of the airfoil. We note that the reason for indenting the contours B and C in the manner shown is to avoid placing the drag contour behind an oblique supersonic shock;

that is, a shock whose downstream flow is supersonic, since the present method uses a first-order accurate hyperbolic operator which tends to dissipate the shock structure over six to ten mesh points. This is an inherent weakness in the present method if sharp supersonic oblique shocks are required.

Subroutine CDCOLE first fixes the locations of the contour boundaries according to the above criteria. In the case of a supersonic free stream, before the calculations are initiated, a check is made to insure that the bow shock is not attached or located too close to the body nose or too close to the upstream boundary. If it is, a message is printed to that effect, the drag computed by a surface pressure integration, and the subroutine exited. If not, the subroutine proceeds to calculate the contour drag integral on the upstream, top, bottom, downstream, and body boundaries, and also on the shocks within the contour using in all cases a trapezoidal rule integration. After the computations are completed, information is printed regarding the drag contribution from each boundary, the total contour drag, the number of shock waves within the contour, the total wave drag, and a message stating whether all the shocks are contained within the contour or whether one or more of the shocks extend beyond the boundaries.

5.2.8 Subroutine CKMESH.- Subroutine CKMESH checks the number of points in both the input x-mesh and the y-mesh. If necessary, the x-mesh is adjusted to contain an odd number of points before the tail and an odd number after the tail, and the y-mesh adjusted to contain an even number of points above and below the airfoil. In counting points in the x-mesh, the point at the trailing edge (I = ITE) is included in both the sets before and after the tail. If the addition of the extra point(s) causes either IMAX or JMAX to exceed 100, a message to that effect is printed, no points are added, and the subroutine exited.

5.2.9 Subroutine CPPLOT.- Subroutine CPPLOT produces a printer plot of the pressure coefficient on the dividing streamline ($y = 0\pm$) ahead of, on, and behind the airfoil. The character "U" is printed for the pressure coefficient on the airfoil upper surface, "L" for the lower surface, "B" for points before and behind the airfoil, and "---" for the critical (sonic) pressure coefficient.

5.2.10 Subroutine CUTOUT.- Subroutine CUTOUT reduces the number of input mesh points to establish a coarse grid for the first attempt at a solution. The x-mesh and the y-mesh are halved, and if possible halved again. If it is desired to use the input mesh to convergence or if the input mesh cannot be refined, the XIN and YIN arrays are loaded into the working arrays X and Y and the routine exited. Otherwise, CUTOUT eliminates half the points in both the x-mesh and y-mesh and adjusts the indices IMAX and JMAX, keeping the first and last points of the meshes at the same location as they were originally.

If it is desired to perform only one grid halving (ICUT = 1), the subroutine loads the remaining points into the X- and Y-arrays and exits. If two mesh cuts are desired (ICUT = 2) and if after the first cut there are an odd number of points before and after the airfoil and an even number below and above the airfoil, the mesh is again halved and the points are loaded into the X- and Y-arrays. After each halving, subroutines ISLIT and JSLIT are called to adjust the I indices at the leading and trailing edges and the J indices immediately above and below the airfoil.

5.2.11 Subroutine DIFCOE.- Subroutine DIFCOE computes on the current (X,Y) mesh the difference coefficients throughout the flow field, including the boundaries, that are required for determining the various finite-difference forms of ϕ_x, ϕ_{xx}, and ϕ_{yy} needed for the solution of the partial differential equation for ϕ. In addition, the difference coefficients required to determine the velocities (ϕ_x, ϕ_y), together with special coefficients needed for extrapolation of flow field properties to the airfoil surface, and special difference coefficients for determining ϕ_{yy} near the airfoil surface are also computed and stored.

5.2.12 Function subroutine DRAG(CDFACT).- Function subroutine DRAG computes the pressure drag by integrating $\phi_x \cdot \phi_y$ around the airfoil surface. The pressure drag is defined as

$$\frac{C_D}{2\delta^{5/3} M_\infty^{-n}} = -\oint_{\text{airfoil}} \phi_x \phi_{\tilde{y}} \, dx$$

or equivalently,

$$\frac{C_D}{2 \cdot \text{CDFACT}} = -\int_0^1 \left[\phi_x^+ \frac{dF_u}{dx} - \phi_x^- \frac{dF_\ell}{dx} \right] dx$$

where $\tilde{y}$ is the transonically scaled y-coordinate, CDFACT $(=\delta^{5/3} M_\infty^{-n})$ is the appropriate transonic scaling factor for the drag coefficient, and

$F_{u,\ell}$ are the affine ordinates describing the airfoil upper and lower surface. The subroutine uses the latter form and computes the integral by a trapezoidal rule integration through subroutine TRAP.

5.2.13 Subroutine DROOTS.- Subroutine DROOTS computes the constants ALPHA0, ALPHA1, ALPHA2, OMEGA0, OMEGA1, and OMEGA2 which are required for the subsonic far-field doublet in an ideal slotted wind tunnel (BCTYPE = 4). The ALPHA's are computed in an iterative fashion as follows:

$$\mathrm{ALPHA0}^n = \frac{\pi}{2} - \tan^{-1}\left[F*\mathrm{ALPHA0}^{n-1} - \frac{\sqrt{K}}{\tilde{P}}\right]$$

$$\mathrm{ALPHA1}^n = \frac{\pi}{2} - \tan^{-1}\left[F*(\mathrm{ALPHA1}^{n-1} - \pi) - \frac{\sqrt{K}}{\tilde{P}}\right]$$

$$\mathrm{ALPHA2}^n = \frac{\pi}{2} - \tan^{-1}\left[F*(\mathrm{ALPHA2}^{n-1} - 2\pi) - \frac{\sqrt{K}}{\tilde{P}}\right]$$

where F is the tunnel slot parameter, $\tilde{P}$ is the transonically scaled wall porosity factor, and K is the transonic similarity parameter. One hundred iterations are allowed for convergence, with the error criterion,

$$|\mathrm{ALPHA}^n - \mathrm{ALPHA}^{n-1}| < 10^{-5}$$

If any ALPHA fails to converge within 100 iterations, an error message is printed and the program stops. If all three alphas converge, then the OMEGA's are computed as follows:

$$\mathrm{OMEGA0} = \left(1 + \frac{F}{1 + \mathrm{ctn}^2(\mathrm{ALPHA0})}\right)^{-1}$$

with analogous formula for OMEGA1 and OMEGA2.

5.2.14 Subroutine ECHINP.- Subroutine ECHINP reads and prints all of the input cards that are used for the runs to follow.

5.2.15 Function subroutine EMACH1(U).- Function subroutine EMACH1 computes either the local Mach number or the transonic similarity parameter based on local velocity. If the input logical variable PHYS = .T., the local Mach number is calculated and set equal to EMACH1 according to $\mathrm{EMACH1(U)} = [1 - (\mathrm{AK} - (\gamma + 1)\mathrm{U})\delta^{2/3}M^{2m}]^{1/2}$ where m depends on the transonic scaling used (see Table 4-I). For PHYS = .F., the local transonic similarity parameter is computed and set equal to EMACH1; that is, EMACH1(U) = AK - $(\gamma + 1)$U.

5.2.16 Subroutine EXTRAP.- Subroutine EXTRAP provides initial values of ϕ for subsonic free-stream flows to subroutine GUESSP at mesh points which lie outside the range of the previous mesh. The formulation is based on the subsonic far-field solution for free air and for various tunnel simulation boundaries (free jet, solid wall, ideal slotted wall, ideal porous/perforated wall). For free air the formulation used is

$$\phi(x,\tilde{y}) = -\frac{\Gamma_{ff}}{2\pi}\cdot\tan^{-1}\left(\frac{\sqrt{K}\tilde{y}}{x}\right) + \frac{D}{2\pi\sqrt{K}}\,\frac{x-\frac{1}{2}}{\left(x-\frac{1}{2}\right)^2 + K\tilde{y}^2}$$

and for tunnel wall boundaries

$$\phi(x,\tilde{y})_{x\to+\infty} = -\frac{\Gamma_{ff}}{2}\left[1 - \operatorname{sgn}(\eta) + (1-J)\psi_o\,\frac{\sin(\eta\beta_o)}{\beta_o}\,e^{-\beta_o\xi}\right]$$

$$+ \frac{D}{2K\tilde{H}}\left[B + \Omega_o\cos(\alpha_o\eta)\,e^{-\alpha_o\xi}\right]$$

$$\phi(x,\tilde{y})_{x\to-\infty} = -\frac{\Gamma_{ff}}{2}\left[1 - \frac{J\tilde{y}}{H(1+F)} - \psi_2\,\frac{\sin[\eta(\pi-\beta_2)]}{\pi-\beta_2}\,e^{(\pi-\beta_2)\xi}\right]$$

$$+ \frac{D}{2K\tilde{H}}\left[(1-B)\Omega_1\cos[(\pi-\alpha_1)\eta]\,e^{(\pi-\alpha_1)\xi}\right]$$

where

$$\xi = \frac{x-\frac{1}{2}}{H\cdot\sqrt{K}}$$

$$\eta = \frac{\tilde{y}}{\tilde{H}}$$

and the other parameters are defined according to the following table.

	B	α_o	α_1	Ω_o	Ω_1	Ω_2	J	β_o	β_1	ψ_o	ψ_2
Solid Wall	$\frac{1}{2}$	π	π	1	1	1	0	$\frac{\pi}{2}$	$\frac{\pi}{2}$	1	1
Free Jet	0	$\frac{\pi}{2}$	$\frac{\pi}{2}$	1	1	1	$\frac{1}{2}$	0	0	1	1
Perforated Wall	0	θ_2	θ_2	1	1	1	0	θ_1	θ_1	1	1
Slotted Wall	0	*	*	*	*	*	$\frac{1}{2}$	*	*	*	*

*No simplication possible for ideal slotted wall--must use formulas below.

$$\cot \alpha_o = F\alpha_o - \theta \qquad 0 \leq \alpha_o \leq \pi$$

$$\cot \alpha_1 = F(\alpha_1 - \pi) - \theta \qquad 0 \leq \alpha_1 \leq \pi$$

$$\Omega_o = \left(1 + \frac{F}{1 + \cot^2\alpha_1}\right)^{-1}$$

$$\Omega_1 = \left(1 + \frac{F}{1 + \cot^2\alpha_1}\right)^{-1}$$

$$\tan \beta_o = -F\beta_o + \theta \qquad 0 \leq \beta_o \leq \frac{\pi}{2}$$

$$\tan \beta_2 = -F(\beta_2 - \pi) + \theta \qquad -\frac{\pi}{2} \leq \beta_2 \leq \frac{\pi}{2}$$

$$\psi_o = \left(1 + \frac{F}{1 + \tan^2\beta_2}\right)^{-1}$$

$$\psi_2 = \left(1 + \frac{F}{1 + \tan^2\beta_o}\right)^{-1}$$

$$\theta = \frac{\sqrt{K}}{\tilde{P}}$$

$$\theta_1 = \tan^{-1}(\theta)$$

$$\theta_2 = \cot^{-1}(\theta)$$

5.2.17 Subroutine FARFLD.- Subroutine FARFLD computes the appropriate boundary data for the outer boundaries of the computational mesh. In the case of a supersonic free stream ($K \leq 0.0$), the upstream boundary conditions correspond to uniform undisturbed flow, the downstream boundary is required to be supersonic, and for the top and bottom boundaries the simple wave solution is employed. In this case, the subroutine simply defines the variable FHINV = 1/F*H and returns.

For a subsonic free stream, the functional form of the potential on the outer boundaries is prescribed and calculated. These forms represent the asymptotic far-field behavior of the potential solution, and are given by a concentrated unit strength doublet and vortex in free air or various wind-tunnel environments, with the doublet and vortex located at $x = 0.5$, $y = 0$. For an ideal slotted wind tunnel (BCTYPE = 4), subroutines DROOTS and VROOTS are called to compute the constants needed in the doublet and vortex solutions; otherwise, the various constants are calculated within the subroutine. We note that the actual boundary values for ϕ are set in subroutines RECIRC and REDUB, where the functional forms determined here are multiplied by the appropriate vortex and doublet strengths.

5.2.18 Subroutine FINDSK(ISTART,IEND,J,ISK).- Subroutine FINDSK identifies the presence of a shock wave on a specified row (J) of the computation mesh between x-points ISTART and IEND. Pairs of ϕ_x values are checked along the row until the condition

$$\text{PX(ISK,J)} \leq \text{SONVEL} \leq \text{PX(ISK-1,J)}$$

is satisfied, where SONVEL is the sonic point streamwise perturbation velocity. If such a pair is found, the subroutine is immediately exited with that value of ISK. If no shock is found, ISK is set equal to -IEND and the subroutine exited.

5.2.19 Subroutine FIXPLT.- Subroutine FIXPLT uses the airfoil surface pressure distribution arrays CPL and CPU to set up the arrays XP, CPUP, CPLO, and CPS for use in subroutine CPPLOT.

5.2.20 Subroutine GUESSP.- Subroutine GUESSP initializes the small-disturbance potential ϕ array as follows:

PSTART = 1, ϕ is set to zero

PSTART = 2, ϕ, X, and Y arrays are read from unit 7 in subroutine READIN together with other flow information about the old solution

PSTART = 3, the arrays from the previous case are already in ϕ, XOLD, YOLD

If PSTART = 2 or 3, the old ϕ array on the XOLD,YOLD mesh must be interpolated onto the current X,Y mesh. A check is made to see if the XOLD mesh is the same as the XIN mesh. If XIN and XOLD are the same mesh, the ϕ array may be interpolated by simple deletion of values at mesh points which have been deleted in subroutine CUTOUT. If the old and new X meshes are not the same, then a simple linear interpolation is made from the old ϕ's providing that each new x-point lies within the range of the old X mesh. If a new x-point lies outside the old X mesh, then for a supersonic free stream ϕ is set to zero, while for a subsonic free stream subroutine EXTRAP is called to extrapolate a new ϕ value using the far-field solution.

The same steps are followed to interpolate ϕ in the y-direction with the following slight difference. If the new y-point lies outside the range of the old Y mesh, then ϕ is set equal to either $\phi_{JMINO,i}$ if the new y value is less than the minimum old y value, or to $\phi_{JMAXO,i}$ if the new y value is greater than the maximum old y value. Only in the case of subsonic free air flow ($K > 0.$, BCTYPE = 1) is subroutine EXTRAP called to extrapolate for ϕ using the far-field formula.

5.2.21 Subroutine INPERR(I).- Subroutine INPERR writes the appropriate error message associated with an error in input and stops the program. There are eight input error messages:

Error Number	Message
1	IMAX or JMAX is greater than 100, not allowed
2	X mesh points not monotonic increasing
3	Y mesh points not monotonic increasing
4	Mach number not in permitted range (0.5,2.0)

Error Number	Message
5	Alpha not in permitted range (-9.0,9.0)
6	Delta not in permitted range (0.0,1.0)
7	AK = 0. Value of AK must be input since PHYS = .F.
8	Mach number too close to 1, not allowed

5.2.22 Subroutine ISLIT(X).- For a given x-mesh array X, subroutine ISLIT computes the I indices associated with the mesh points on the airfoil surface ($0. \leq x \leq 1.$) at or just behind the leading edge (I = ILE) and at or just ahead of the trailing edge (I = ITE).

5.2.23 Subroutine JSLIT(Y).- For a given y-mesh array Y, subroutine JSLIT computes the J indices associated with the rows of mesh points immediately below (J = JLOW) and above (J = JUP) the airfoil ($y = 0.$).

5.2.24 Function LIFT (CLFACT).- The real function LIFT computes the lift coefficient from the jump in potential $\Delta\phi$ at the trailing edge. The lift coefficient is defined as

$$C_L = -\int_0^1 (C_{p_u} - C_{p_\ell})dx = +2\delta^{2/3}M_\infty^{-n}\int_0^1 [(\phi_x)_u - (\phi_{x_\ell})]dx$$

or, equivalent, from an integration by parts

$$C_L = 2\cdot\text{CLFACT}\cdot\Delta\phi_{te}$$

where CLFACT ($=\delta^{2/3}M_\infty^{-n}$) is the appropriate transonic scaling factor for the lift coefficient.

5.2.25 Subroutine MACHMP.- Subroutine MACHMP produces a print map of the local Mach numbers, rounded to the nearest 0.1, throughout the flow field. The row of points immediately above the airfoil are designated by (+), the row immediately below by (-), the leading edge by (L), and the trailing edge by (T). Supersonic points are designated by an asterisk (*).

5.2.26 Subroutine MILINE.- Subroutine MILINE determines the coordinates where sonic velocity occurs and outputs them. Beginning with row J = JMAX, pairs of ϕ_x values are checked along the row to determine whether either of the two conditions

$$PX(I,J) \leq SONVEL \leq PX(I-1,J)$$

$$PX(I,J) \geq SONVEL \geq PX(I-1,J)$$

is satisfied, where SONVEL is the streamwise perturbation velocity at a sonic point. If either of these conditions are met, a linear interpolation is made between X(I,J) and X(I - 1,J), that computed x-coordinate stored temporarily in array XSONIC and permanently in array XSLPRT, and the y-coordinate loaded into array YSLPRT. After each row is scanned, the XSONIC array and the J index of the row are printed. When the row immediately under the airfoil is reached, the heading "BODY LOCATION" is printed if at least one sonic point has been found.

If more than 200 sonic points are found, a message to this effect is printed and the subroutine is exited. After the entire grid has been checked for sonic points and if at least one has been found, the subroutine checks whether any of the sonic points lie on the outer boundaries. If one does and if the case under study is either a subsonic free-stream calculation or a free air (BCTYPE = 1) supersonic free-stream calculation, the following message is printed out:

```
                    * * * * * CAUTION * * * * *

      SONIC LINE HAS REACHED A BOUNDARY
      THIS VIOLATES ASSUMPTIONS USED TO DERIVE BOUNDARY CONDITIONS
      SOLUTION IS PROBABLY INVALID
```

Finally, the subroutine defines the boundaries and step increments for the printer plot of the sonic line and passes these constants along with the arrays XSLPRT and YSLPRT and the number NPTS of sonic points into a call to the subroutine PLTSON, which plots the sonic line.

5.2.27 Subroutine NEWISK (ISKOLD,J,ISKNEW).- Subroutine NEWISK locates an updated x-position (I = ISKNEW) of the shock wave on a specified row J given an initial guess for the location (I = ISKOLD). The subroutine searches the range $ISKOLD - 3 \leq I \leq ISKOLD + 2$ testing the condition $PX(ISKNEW - 1,J) \geq SONVEL \geq PX(ISKNEW,J)$. If the condition is met, the subroutine returns with the current value of ISKNEW. Otherwise, with no shock being found, ISKNEW is set negative and the subroutine exited.

5.2.28 Function PITCH (CMFACT).- Function PITCH computes the airfoil pitching moment (positive nose-up) about $x = x_m$, $y = 0$. Pitching moment is defined as

$$C_m = \int_0^1 (x - x_m)(C_{p_u} - C_{p_\ell})\,dx$$

or, equivalently

$$C_m = -2\cdot \mathrm{CMFACT}\cdot (1 - x_m)\cdot \Delta\phi_{te} - \int_0^1 \Delta\phi \; dx$$

where CMFACT (= $\delta^{2/3}M_\infty^{-n}$) is the appropriate transonic scaling for the moment coefficient. The subroutine uses the latter form for C_m and computes the second integral by trapezoidal rule integration through subroutine TRAP. We note that the subroutine sets $x_m = 0.25$, so that the pitching moment is calculated about the quarter chord.

5.2.29 Subroutine PLTSON (X,Y,XAXMIN,XMAX,XINCR,YAXMIN,YMAX,YINCR, NPTS).- Subroutine PLTSON is a modified version of a printer plot routine developed by M. S. Itzkowitz, May 1967. The subroutine provides a printer plot of the sonic lines that are present in the flow field. The plot is provided on a 51 × 101 point rectangular grid with $-1.0 \leq y \leq 1.5$ and $0.75 \leq x \leq 1.75$, and uses 56 printer lines. The argument list is all input and is defined as follows:

X	array of x-ordinates to be plotted
Y	array of y-ordinates to be plotted
XAXMIN	minimum x (left) grid boundary
XMAX	maximum x (right) grid boundary
XINCR	increment between x-grid marks
YAXMIN	minimum y (lower) grid boundary
YMAX	maximum y (upper) grid boundary
YINCR	increment between y-grid marks
NPTS	dimension of (X,Y) arrays

If either of the incremental step sizes is zero, the program will exit with no plot produced. The input arrays are not destroyed during the calculation.

5.2.30 Subroutine PRBODY.- Subroutine PRBODY prints out the airfoil geometrical characteristics. If logical variable PHYS = .T., all dimensions are normalized by the airfoil chord; otherwise, all dimensions except x are normalized by chord length times thickness ratio. Quantities printed are maximum thickness, volume of airfoil per unit span, maximum camber, and for each input x-mesh point along the airfoil, the y-ordinates of the upper and lower surfaces of the airfoil, the thickness distribution, and camberline.

5.2.31 Subroutine PRINT.- Subroutine PRINT is the main print control of program output. The subroutine prints header information regarding:

(i) choice of printout in similarity on physical variables

(ii) definition of transonic scaling (Cole, Spreiter, Krupp, or user)

(iii) boundary condition type (free air, free jet, solid wall, slotted wall, or porous wall)

(iv) choice of differencing (fully conservative or not conservative) at shock

(v) choice of Kutta condition or lift specification

and also outputs the Mach number, thickness ratio, angle of attack, transonic similarity parameter, far-field doublet strength, airfoil volume (per unit span), and the values of the transonic scaling parameters used for scaling the coefficients of pressure, lift, drag, and moment, the y-coordinate, and the vertical velocity. The rest of the print output is performed by calling the following subroutines: PRINT1, FIXPLT, PRTWAL, MILINE, PRTFLD, and CDCOLE.

5.2.32 Subroutine PRINT1.- Subroutine PRINT1 outputs the pressure coefficient and local Mach number (or local similarity parameter if PHYS = .F.) on the airfoil and dividing streamline ($y = 0.\pm$) at each x-point of the current mesh, and also provides a printer plot of the pressure coefficient along side the tabulated values. In addition, the y-coordinate array Y is printed as well as the lift, moment, and sonic pressure coefficients. Since PRINT1 is used to putput these quantities on each grid (coarse, medium, and fine), a header is written to identify the current grid. If the local Mach number exceeds 1.3, a warning message is printed indicating that the predicted shock jumps may be in error. Also, if a detached bow shock wave is between IMIN and IMIN + 1, a message is printed indicating that the detached shock will lie upstream of the current x mesh, the ABORT parameter set equal to true and the calculation terminated.

5.2.33 Subroutine PRTFLD.- Subroutine PRTFLD outputs the pressure coefficient, flow angle, and Mach number in the flow field. The number of rows (J lines) printed is set by the user-input value of PRTFLO as follows:

PRTFLO = 1, no J lines are printed
PRTFLO = 2, all J lines are printed
PRTFLO = 3, three J lines around the row where the maximum iteration error in ϕ occurs (i.e., J = JERROR) are printed

5.2.34 Subroutine PRTMC.- Subroutine PRTMC produces a local flow character map of the flow field by ascribing to each point in the input mesh a letter designating whether the flow at that point is elliptic (subsonic), parabolic (sonic), hyperbolic (supersonic), or shock (shock point) according to the convention:

-	Elliptic	(Subsonic)
P	Parabolic	(Sonic)
H	Hyperbolic	(Supersonic)
S	Shock Point	(Shock)

5.2.35 Subroutine PRTSK (Z,ARG,L,NSHOCK,CDSK,LPRT1).- Subroutine PRTSK provides an inviscid wake profile for all of the shock waves contained within the momentum contour used to calculate the drag. The wave drag contribution CD(y) and total pressure loss $P_o(y)/P_\infty$ along each shock wave is given as a function of the transonically scaled y-coordinate. In addition, the total wave drag for each shock is also provided. If a shock wave extends outside the contour, a message to that effect is printed indicating that shock losses are not available for that portion of the shock.

5.2.36 Subroutine PRTWAL.- Subroutine PRTWAL determines the pressure coefficient and the flow angle on the wind-tunnel walls ($y = \pm H$) according to the particular type of wall, prints that information, and also provides a printer plot of the pressure distribution along side the tabulated printout. Information regarding the wall type, tunnel half height, wall porosity factor (if applicable), tunnel slot parameter (if applicable), and critical pressure coefficient are also provided.

5.2.37 Function PX(I,J).- Function PX computes the streamwise perturbation velocity ϕ_x at the point (I,J). Different differencing forms are used depending whether the point is on the upstream or downstream boundaries or is an interior point. For the interior points, ϕ_x is given by

$$\phi_x = \frac{1}{2}\left[\frac{\phi_{j,i+1} - \phi_{j,i}}{x_{i+1} - x_i} + \frac{\phi_{j,i} - \phi_{j,i-1}}{x_i - x_{i-1}}\right]$$

while on the upstream boundary (I = IMIN)

$$\phi_x = \frac{3}{2}\left[\frac{\phi_{j,i+1} - \phi_{j,i}}{x_{i+1} - x_i}\right] - \frac{1}{2}\left[\frac{\phi_{j,i+2} - \phi_{j,i+1}}{x_{i+2} - x_{i+1}}\right]$$

and on the downstream boundary (I = IMAX)

$$\phi_x = \frac{3}{2}\left[\frac{\phi_{j,i} - \phi_{j,i-1}}{x_i - x_{i-1}}\right] - \frac{1}{2}\left[\frac{\phi_{j,i-1} - \phi_{j,i-2}}{x_{i-1} - x_{i-2}}\right]$$

5.2.38 Function PY (I,J).- Function PY computes the vertical perturbation velocity ϕ_y at the point (I,J). Different differencing formulas are employed depending whether the point is on the upper or lower boundaries or on the rows immediately above or below the airfoil. In the latter case, separate formulations are involved depending upon whether the point is ahead, over or under, or behind the airfoil. For all of the other points,

$$\phi_y = \frac{1}{2}\left[\frac{\phi_{j+1,i} - \phi_{j,i}}{y_{j+1} - y_j} + \frac{\phi_{j,i} - \phi_{j-1,i}}{y_j - y_{j-1}}\right]$$

The exceptional cases are:

Lower Boundary (j = JBOT)

$$\phi_y = \frac{3}{2}\left[\frac{\phi_{j+1,i} - \phi_{j,i}}{y_{j+1} - y_j}\right] - \frac{1}{2}\left[\frac{\phi_{j+2,i} - \phi_{j+1,i}}{y_{j+2} - y_{j+1}}\right]$$

Upper Boundary (j = JTOP)

$$\phi_y = \frac{3}{2}\left[\frac{\phi_{j,i} - \phi_{j-1,i}}{y_j - y_{j-1}}\right] - \frac{1}{2}\left[\frac{\phi_{j-1,i} - \phi_{j-2,i}}{y_{j-1} - y_{j-2}}\right]$$

Row below airfoil (j = JLOW)

(a) Ahead

$$\phi_y = \frac{1}{2}\left[\frac{\phi_{JUP,i} - \phi_{JLOW,i}}{y_{JUP} - y_{JLOW}} + \frac{\phi_{JLOW,i} - \phi_{JLOW-1,i}}{y_{JLOW} - y_{JLOW-1}}\right]$$

(b) Under

$$\phi_y = \frac{1}{2}\left[\left(\frac{dF_\ell}{dx}\right)_i - \alpha + \frac{\phi_{JLOW,i} - \phi_{JLOW-1,i}}{y_{JLOW} - y_{JLOW-1}}\right]$$

(c) Behind

$$\phi_y = \frac{1}{2}\left[\frac{\phi_{JUP,i} - \Delta\phi_i - \phi_{JLOW,i}}{y_{JUP} - y_{JLOW}} + \frac{\phi_{JLOW,i} - \phi_{JLOW-1,i}}{y_{JLOW} - y_{JLOW-1}}\right]$$

Row above airfoil (j = JUP)

(a) Ahead

$$\phi_y = \frac{1}{2}\left[\frac{\phi_{JUP,i} - \phi_{JLOW,i}}{y_{JUP} - y_{JLOW}} + \frac{\phi_{JUP+1,i} - \phi_{JUP,i}}{y_{JUP+1} - y_{JUP}}\right]$$

(b) Over

$$\phi_y = \frac{1}{2}\left[\left(\frac{dF_u}{dx}\right)_i - \alpha + \frac{\phi_{JUP+1,i} - \phi_{JUP,i}}{y_{JUP+1} - y_{JUP}}\right]$$

(c) Behind

$$\phi_y = \frac{1}{2}\left[\frac{\phi_{JUP,i} - \Delta\phi_i - \phi_{JLOW,i}}{y_{JUP} - y_{JLOW}} + \frac{\phi_{JUP+1,i} - \phi_{JUP}}{y_{JUP+1} - y_{JUP}}\right]$$

5.2.39 Subroutine READIN.- All user-option input to the program is read in subroutine READIN. The subroutine first calculates and prints the elapsed time in seconds that the previous case took to run, or for the first case sets the start time. Next, a check is made to determine if any test cases are left. If there are and if the user has opted to use the previous result for ϕ as a first guess for the current case (PSTART = 3), then a test is made to determine if the ϕ array in core from the previous case is usable (i.e., if the previous calculation was not aborted). Next, if the default grids for either subsonic free air (XKRUPP,YFREE) or tunnel (XKRUPP,YTUN), or the analytic default mesh (AYMESH = TRUE) are to be used, the subroutine loads those grids into the XIN, YIN arrays and then prints out all of the user-option input parameters, including those having default values. A check is then made to insure that the input mesh does not exceed 100 points in either direction and that the mesh ordinates are monotonic increasing. Next, checks are made on M_∞, α, and δ to insure they are within the permitted range, and then the indices of the leading

(I = ILE) and trailing (I = ITE) edges and the rows immediately above (J = JUP) and below (J = JLOW) the airfoil are determined. Subroutine CKMESH is then called to insure that certain grid areas have an appropriate (odd or even) number of points. If the case underway is a tunnel simulation calculation, the bounds of the grid are checked and adjusted if necessary. Finally, if the initial guess for the ϕ array is to be read from unit 7 (PSTART = 2), the necessary input from that calculation is read and then printed.

5.2.40 Subroutine RECIRC.- Subroutine RECIRC computes the jump in ϕ at the trailing edge, $\Delta\phi_{te}$, the updated far-field circulation, Γ_{ff}^{n+1}, and the jump in ϕ, $\Delta\phi(x)$, along the slit $y = 0.$, $x > 1.$, according to the following formulas.

For KUTTA = .T.,

$$\Delta\phi_{te} = \phi_{te}^{+} - \phi_{te}^{-} = \frac{\phi_{JUP,ITE} \cdot y_{JUP+1} - \phi_{JUP+1,ITE} \cdot y_{JUP}}{y_{JUP+1} - y_{JUP}}$$

$$- \frac{\phi_{JLOW-1,ITE} \cdot y_{JLOW} - \phi_{JLOW,ITE} \cdot y_{JLOW-1}}{y_{JLOW} - y_{JLOW-1}}$$

$$\Gamma_{ff}^{n+1} = (1. - \omega_{\Gamma}) \cdot \Gamma_{ff}^{n} + \omega_{\Gamma} \Delta\phi_{te}$$

$$\Delta\phi(x) = \Delta\phi_{te} + \frac{x - 1}{x_{max} - 1} \left(\Gamma_{ff}^{n+1} - \Delta\phi_{te}\right)$$

where ω_{Γ} is the relaxation factor for circulation, while for KUTTA = .F.,

$$\Delta\phi_{te} = \frac{CLSET}{2}$$

$$\Gamma_{ff} = \frac{CLSET}{2}$$

$$\Delta\phi(x) = \frac{CLSET}{2}$$

where CLSET is the specified lift coefficient.

5.2.41 Subroutine REDUB.- Subroutine REDUB computes the far-field doublet strength. For lifting free air flows (BCTYPE = 1 and ABS(CIRCFF) $> 10^{-4}$), the doublet strength is set equal to the model volume. For other flows the nonlinear contribution (DBLSUM = $((\gamma + 1)/4) \cdot \iint \phi_x^2 dx\, dy$) is added to the volume. The double integral is computed by considering ϕ_x to be centered midway between mesh points; that is,

$$\text{DBLSUM} = \frac{\gamma + 1}{4} \sum_{i=\text{IMIN}}^{\text{IMAX}} \Delta x_{i+1/2} \int_{y_{\text{JMIN}}}^{y_{\text{JMAX}}} (\phi_x^2)_{i+1/2} dy$$

so that

$$\text{DBLSUM} = \frac{\gamma + 1}{4} \sum_{i=\text{IMIN}}^{\text{IMAX}} \frac{1}{x_{i+1} - x_i} \int_{y_{\text{JMIN}}}^{y_{\text{JMAX}}} (\phi_{j,i+1} - \phi_{j,i})^2 dy$$

and the y-integration is performed using a modified trapezoidal rule.

We note that the doublet strength is required only for $M_\infty < 1$. For subsonic free air lifting flows, the doublet integral DBLSUM is not convergent since $\phi \sim \Gamma_{ff} \tan^{-1}(y/x)$; in this case, a more complicated integral applies (see Krupp, ref. 3). This program neglects that contribution, and while this procedure introduces an error, numerical experimentation has shown it not to be significant.

5.2.42 Subroutine REFINE.- Subroutine REFINE doubles the number of mesh points in both the x- and y-meshes by inserting new mesh points halfway between the old mesh points. The subroutine also linearly interpolates the ϕ array to the new points, determines the new (I,J) indices of the leading (ILE) and trailing (ITE) edges, and the rows immediately above (JUP) and below (JLOW) the airfoil by calls to subroutines ISLIT and JSLIT, respectively, and uses a linear extrapolation to calculate ϕ on the new JLOW and JUP rows.

5.2.43 Subroutine RESET.- Subroutine RESET updates the far-field boundary conditions for subsonic free-stream flows. The potential ϕ is recalculated on the upstream (IMIN) and downstream (IMAX) boundaries for both free air or wind-tunnel simulations. The update is accomplished at the end of each iteration sweep through the flow field. For free air flows (BCTYPE = 1), new boundary values are also calculated on the top (JMAX) and bottom (JMIN) boundaries; while for wind-tunnel simulations, this calculation is omitted.

5.2.44 Subroutine SAVEP.- Subroutine SAVEP automatically stores restart data from the currently completed calculation in an "old data" block. The subroutine first rescales the parameters back into the input format and then stores the following parameters and arrays: TITLE(I), IMIN, JMIN, IMAX, JMAX, CL, EMACH, ALPHA, DELTA, VOL, DUB, X(I), and YIN(J). If the user has opted to save the restart block (PSAVE = .T.), then the subroutine also writes the above information on Tape 3.

5.2.45 Subroutine SCALE.- Subroutine SCALE scales physical variables that are required for program computation. The option to scale or not is controlled by the logical input variable PHYS. For PHYS = .T., all input and output quantities are in physical units normalized by free-stream values and airfoil chord, and SCALE reduces them to transonic variables according to the following convention:

SIMDEF = 1, Cole scaling
SIMDEF = 2, Spreiter scaling
SIMDEF = 3, Krupp scaling
SIMDEF = 4, User-option scaling; if this option is used, the definition of local Mach number must also be adjusted in EMACH1

For PHYS = .F., input is already in scaled variables and no further scaling is done. In this case, the user must input the value of the transonic similarity parameter AK $= (1 - M_\infty^2)/M_\infty^{2m}\delta^{2/3}$ where the exponent m is now user-specified.

Subroutine SCALE then checks to determine that AK has an appropriate value for the case being considered, computes the square root of AK(RTK), the sonic velocity (SONVEL), and the critical pressure coefficient (CPSTAR) and then exits.

5.2.46 Subroutine SETBC.- Subroutine SETBC sets the limits on the range of the I and J indices for solution of the difference equations according to the following schedule:

$M_\infty < 1$				
BCTYPE	IUP	IDOWN	JTOP	JBOT
1	IMIN+1	IMAX-1	JMAX-1	JMIN+1
2	IMIN+1	IMAX-1	JMAX	JMIN
3	IMIN+1	IMAX-1	JMAX-1	JMIN+1
4	IMIN+1	IMAX-1	JMAX	JMIN
5(POR $\leq$ 1.5)	IMIN+1	IMAX-1	JMAX	JMIN
5(POR $>$ 1.5)	IMIN+1	IMAX-1	JMAX-1	JMIN+1

$M_\infty > 1$				
BCTYPE	IUP	IDOWN	JTOP	JBOT
1	IMIN+2	IMAX	JMAX	JMIN
2	IMIN+2	IMAX	JMAX	JMIN
3	IMIN+2	IMAX	JMAX-1	JMIN+1
4	IMIN+2	IMAX	JMAX	JMIN
5(POR $\leq$ 1.5)	IMIN+2	IMAX	JMAX	JMIN
5(POR $>$ 1.5)	IMIN+2	IMAX	JMAX-1	JMIN+1

The subroutine also computes the body slope boundary conditions at the current x-mesh points on the body, multiplies them by the appropriate mesh spacing constants and stores them into arrays FXUBC(J), FXLBC(I) for use in solving the finite-difference equations at points on the rows immediately above and below the airfoil

5.2.47 Subroutine SIMP (R,X,Y,N,IER).- Subroutine SIMP calculates the integral $\int y\,dx$ by using Simpson's rule integration. The subroutine argument list is defined as follows:

R	resultant value of the integral
X	vector array of x points
Y	vector array of y values

N length of vector arrays (X,Y)

IER error control index; equal to 1 for satisfactory result; equal to 2 when only one point is in the interval (N = 1); equal to 4 when X array not monotonic

Equal point spacing is not required and the number of points N can be either odd or even. In the latter case, a polynomial fit through the first three points is used and an analytical integration performed from X(1) to X(2) in order to obtain an odd number of points for the remaining interval, X(2) to X(N).

5.2.48 Subroutine SOLVE.- Subroutine SOLVE controls the sequence of calculations involved in the finite-difference solution for ϕ. The potential is calculated by using a successive line over-relaxation (SLOR) algorithm and sweeping the flow field from upstream to downstream limits one column at a time. At the start of each sweep, subroutine RECIRC is called to compute the jump in potential at the trailing edge, the circulation for the far-field boundary, and the jump in potential along the slit $y = 0$, $x > 1$. Next, subroutine SYOR is called to compute a new value of the potential at all the grid points. For subsonic flows, after the program has completed an iteration sweep, and if the current calculation is a free air case, the potential is updated throughout the flow field by adding the increment in potential due to the change in circulation multiplied by the far-field vortex behavior. Then subroutine REDUB is called (but only at every NDUBth iteration sweep) to recompute the far-field doublet strength, and then subroutine RESET is called (at each iteration) to update the far-field boundary conditions. At every IPRTERth iteration, the iteration count (ITERATION), lift coefficient (CL), moment coefficient (CM), I and J locations (IERR,JERR) where the maximum iteration error in ϕ occurs, the absolute magnitude of the maximum iteration error in ϕ (ERROR), the I and J locations (IRL,JRL) of the maximum residual, the absolute magnitude of the maximum residual (BIGRL), the absolute magnitude of the iteration error in circulation (ERCIRC), the I location (ICPU) of the point on the airfoil where the maximum iteration error in upper surface pressure coefficient occurs, the absolute magnitude of the maximum iteration error in upper surface pressure coefficient (CPERRU), and the corresponding results (ICPL,CPERRL) for the maximum iteration error in surface pressure coefficient. The results are then checked for convergence of potential. If the test is met, the message "SOLUTION CONVERGED" is printed and the subroutine exited. If the convergence criteria is not met, a divergence test on ϕ is performed. If the maximum iteration error in ϕ has become greater than the divergence control (DVERGE), the error message "SOLUTION DIVERGED" is printed and the calculation aborted. If the solution has neither converged nor diverged, the iteration sweeps are continued up to the limit, MAXIT, at which point the message "ITERATION LIMIT REACHED" is printed and the subroutine exited.

5.2.49 Subroutine SPLN1 (X,Y,N).- Subroutine SPLN1 is a cubic spline interpolation subroutine which determines both the ordinate and slope of a function using a cubic spline fit. The argument list consists of a table of (X,Y) ordinates and is defined as follows:

X	array of independent variable
Y	array of dependent variable
N	dimension of (X,Y) arrays

There are two entry points to this subroutine. A call to SPLN1 initializes the cubic spline coefficient arrays, while a call to SPLN1X will determine the spline-fit ordinate (YP) and slope (DYP) at the specified point XP. We note that XP,YP, and DXP are passed through a name common block (SPLN) rather than an argument list.

To use SPLN1 it is necessary to specify a derivative of Y at the end points X(1),X(N). Either the first or second derivative at these points may be given. Control for this option is through the parameters (K1,K2) according to the following:

K1 = 1, DY1 = first derivative of Y at X(1)
K1 = 2, DY1 = second derivative of Y at X(1)
K2 = 1, DY2 = first derivative of Y at X(N)
K2 = 2, DY2 = second derivative of Y at X(N)

Values for (K1,K2,DY1,DY2) must be specified prior to the call to SPLN1.

5.2.50 Subroutine SYOR.- Subroutine SYOR determines the potential ϕ at the interior grid points throught the flow field; that is, JBOT $\le$ J $\le$ JTOP, IUP $\le$ I $\le$ IDOWN. This is accomplished by solving the tridiagonal matrix equation $A\overline{C}_i = f_i$ where $\overline{C}_i$ represents the correction potential and the N dimensional column vector (with N = JTOP - JBOT + 1) defined by

$$\bar{c}_i = \begin{pmatrix} c_{JBOT,i} \\ \vdots \\ c_{JTOP,i} \end{pmatrix}$$

and where A is an NXN dimensional, diagonally-dominant, tridiagonal matrix, and f_i is an N dimensional column vector.

The following solution sequence is repeated for each x-station from IUP to IDOWN. The coefficient $1/(x_i - x_{i-1})^2$ of the ϕ_{xt} term is calculated. Then the array VC_j representing the coefficient of ϕ_{xx} in the partial differential equation; that is, $(K - (\gamma + 1)\phi_x)/2\Delta x$, is computed at each J station. Next, the arrays representing the diagonal $(DIAG_j)$, subdiagonal (SUB_j), and superdiagonal (SUP_j) elements of the tridiagonal matrix A are determined, where account is taken in calculating $DIAG_j$ of whether VC_j is greater or less than zero. The array RHS which is equal to the negative of the residual $(-R_{ij})$, and which represents the right-hand side f_i of the matrix equation is then computed. If the iteration sweep in x has proceeded to the airfoil; that is, ILE $\leq$ I $\leq$ ITE, the $DIAG_j$, SUB_j, SUP_j, and RHS_j arrays are modified at the points immediately below (j = JLOW) and above (j = JUP) the x axis to account for the airfoil boundary condition. A similar procedure is used at (j = JLOW,JUP) at locations behind the trailing edge to account for the Kutta condition jump in potential along the cut extending from the trailing edge to the downstream boundary. Next subroutine BCEND is called to modify the diagonal $(DIAG_j)$ and right-hand-side (RHS_j) arrays to account for the appropriate boundary conditions at JBOT and JTOP. The current column is then scanned to see whether the maximum residual $|R_{ij}|_{max} = |RHS_j|$ is contained within it. If it is, that value (BIGRL) and the I,J) locations (IRL,JRL) are stored. Finally, modifications are made to $DIAG_j$ and RHS_j to add the $\epsilon \Delta t \phi_{xt}/\Delta x$ term according to

$$DIAG_j = DIAG_j - \frac{\epsilon}{(x_i - x_{i-1})^2}$$

$$RHS_j = RHS_j - \epsilon \frac{\left(\phi^{(n+1)}_{j,i-1} - \phi^{(n)}_{j,i-1}\right)}{(x_i - x_{i-1})^2}$$

The tridiagonal matrix equation $A\,\overline{C}_i = f_i$ is then solved by using the method of triangular decomposition. The NXN dimensional, tridiagonal matrix A is factored into the product $A = LU$, where L is a lower triangular matrix containing only subdiagonal and diagonal elements and U is a special upper triangular matrix containing only superdiagonal elements and unit diagonal elements. Thus, the original equation

$$A\overline{C}_i = L\,U\,\overline{C}_i = f_i$$

is factored into two equations

$$U\,\overline{C}_i = Z$$

$$L\,Z = f_i$$

The latter equation is solved first for Z and then the former equation is solved for $\overline{C}_i$ by back-substitution.

Finally, the new potential is computed by adding the correction to the old potential, and the correction potential $\overline{C}_i$ vector is scanned to determine whether the maximum correction is contained within it. If it is, that value (ERROR) and the (I,J) location (IERROR, JERROR) are stored for output.

5.2.51 Subroutine TRAP (X,Y,N,SUM).- Subroutine TRAP performs an integration of the vector array Y with respect to the ordinates in the vector array X by using the trapezoidal rule. Here, N is the length of the X,Y arrays and SUM is the resultant integral. The formula used is

$$\text{SUM} = \frac{1}{2}\sum_{i=1}^{N-1}(x_{i+1} - x_i)(y_{i+1} + y_i)$$

5.2.52 Main program TSFOIL (INPUT, OUTPUT, TAPE5 = INPUT, TAPE6 = OUTPUT, TAPE3, TAPE7).- TSFOIL is the main program. No calculations are performed within this routine, which acts only to call the necessary subroutines to read input, perform the required calculations, and print the output. Two tests are made within the routine that determine: (1) when the final computation mesh has been reached; and (2) whether the current calculation has been aborted. If either of these tests are positive, the program prints the final output for the current case, resets the mesh parameters back to those for the input mesh, terminates the current calculation, and goes on to read the input for the following case.

5.2.53 Subroutine VROOTS.- Subroutine VROOTS computes the constants BETA0, BETA1, BETA2, PSI0, PSI1, and PSI2 which are required for the subsonic far-field vortex in an ideal slotted wind tunnel (BCTYPE = 4). The BETA's are computed in an iterative fashion as follows:

$$\mathrm{BETA0}^{n} = \tan^{-1}\left[-F\cdot \mathrm{BETA0}^{n-1} + \frac{\sqrt{K}}{\tilde{P}}\right]$$

$$\mathrm{BETA1}^{n} = \tan^{-1}\left[-F\cdot (\mathrm{BETA1}^{n-1} + \pi) + \frac{\sqrt{K}}{\tilde{P}}\right]$$

$$\mathrm{BETA2}^{n} = \tan^{-1}\left[-F\cdot (\mathrm{BETA2}^{n-1} - \pi) + \frac{\sqrt{K}}{\tilde{P}}\right]$$

where F is the tunnel slot parameter, K the transonic similarity parameter, and $\tilde{P}$ the transonically scaled wall porosity factor. One hundred iterations are allowed for convergence with the error criteria

$$|\mathrm{BETA}^{n} - \mathrm{BETA}^{n-1}| < 10^{-5}$$

If any BETA fails to converge within 100 iterations, an error message is printed and the program stops. If all three BETA's converge, then the PSI's are computed as follows:

$$\mathrm{PSI0} = 1 + \frac{F}{1 + \tan^{2}(\mathrm{BETA0})}$$

with analogous formulas for PSI1 and PSI2.

APPENDIX
LISTING OF COMPUTER PROGRAM TSFOIL

```
      PROGRAM TSFOIL(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE3,TAPE7)        TSFOIL   2
C**************************************************************************TSFOIL   3
C                                                                          TSFOIL   4
C                 MAIN PROGRAM FOR TSFOIL                                  TSFOIL   5
C                 PROGRAM COMPUTES TRANSONIC FLOW PAST A TWODIMENSIONAL    TSFOIL   6
C                 LIFTING AIRFOIL USING TRANSONIC SMALL DISTURBANCE THEORY TSFOIL   7
C                                                                          TSFOIL   8
C                 PROGRAM WRITTEN BY                                       TSFOIL   9
C                    EARLL M. MURMAN AND FRANK R. BAILEY                   TSFOIL  10
C                    NASA-AMES RESEARCH CENTER                             TSFOIL  11
C                         AND                                              TSFOIL  12
C                    MARGARET L. JOHNSON                                   TSFOIL  13
C                    COMPUTER SCIENCES CORPORATION                         TSFOIL  14
C                                                                          TSFOIL  15
C                DOCUMENTED BY                                             TSFOIL  16
C                    STEPHEN S. STAHARA                                    TSFOIL  17
C                    NIELSEN ENGINEERING AND RESEARCH, INC.                TSFOIL  18
C**************************************************************************TSFOIL  19
      LOGICAL          ABORT                                               COM3     2
      COMMON / COM3/ IREF      , ABORT    , ICUT     , KSTEP               COM3     3
C                                                                          TSFOIL  21
C                 WRITE HEADER PAGE INFORMATION                            TSFOIL  22
C                                                                          TSFOIL  23
      WRITE (6,1000)                                                       TSFOIL  24
      WRITE (6,1001)                                                       TSFOIL  25
      WRITE (6,1002)                                                       TSFOIL  26
      WRITE (6,1003)                                                       TSFOIL  27
      WRITE (6,1002)                                                       TSFOIL  28
      WRITE (6,1004)                                                       TSFOIL  29
      WRITE (6,1002)                                                       TSFOIL  30
      WRITE (6,1005)                                                       TSFOIL  31
      WRITE (6,1002)                                                       TSFOIL  32
      WRITE(6,1006)                                                        TSFOIL  33
      WRITE(6,1002)                                                        TSFOIL  34
      WRITE(6,1007)                                                        TSFOIL  35
      WRITE(6,1002)                                                        TSFOIL  36
      WRITE (6,1001)                                                       TSFOIL  37
 1000 FORMAT(1H1/////////////////////)                                     TSFOIL  38
 1001 FORMAT(24X,69(1H*))                                                  TSFOIL  39
 1002 FORMAT(24X,1H*,67X,1H*)                                              TSFOIL  40
 1003 FORMAT(24X,1H*,25X,15H PROGRAM TSFOIL,27X,1H*/                       TSFOIL  41
     1  24X,1H*,29X, 7H SOLVES,31X,1H*/                                    TSFOIL  42
     2  24X,1H*, 5X,56H INVISCID FLOW PAST THIN TWO DIMENSIONAL LIFTING    TSFOIL  43
     3AIRFOIL,6X,1H*/                                                      TSFOIL  44
     4  24X,1H*,29X, 6H USING,32X,1H*/                                     TSFOIL  45
     5  24X,1H*,15X,35H TRANSONIC SMALL DISTURBANCE THEORY,17X,1H*/        TSFOIL  46
     6  24X,1H*, 9X,47H FULLY CONSERVATIVE FINITE DIFFERENCE EQUATIONS,    TSFOIL  47
     7  11X,1H*/                                                           TSFOIL  48
     8  24X,1H*,17X,31H SUCCESSIVE LINE OVERRELAXATION,19X,1H*)            TSFOIL  49
 1004 FORMAT(24X,1H*,27X,11H WRITTEN BY,29X,1H*)                           TSFOIL  50
 1005 FORMAT(24X,1H*,15X,36H EARLL M. MURMAN AND FRANK R. BAILEY,16X,1H*   TSFOIL  51
     1        24X,1H*,19X,26H NASA-AMES RESEARCH CENTER,22X,1H*/           TSFOIL  52
     2        24X,1H*,19X,26H MOFFETT FIELD, CALIFORNIA,22X,1H*/           TSFOIL  53
     3        24X,1H*,31X, 4H AND,32X,1H*/                                 TSFOIL  54
     4        24X,1H*,23X,20H MARGARET L. JOHNSON,24X,1H*/                 TSFOIL  55
     5        24X,1H*,18X,30H COMPUTER SCIENCES CORPORATION,19X,1H*/       TSFOIL  56
     6        24X,1H*,20X,26H MOUNTAIN VIEW, CALIFORNIA,21X,1H*)           TSFOIL  57
 1006 FORMAT(24X,1H*,25X,14H DOCUMENTED BY,28X,1H*)                        TSFOIL  58
 1007 FORMAT(24X,1H*,23X,19H STEPHEN S. STAHARA,25X,1H*/                   TSFOIL  59
     1   24X,1H*,14X,39H NIELSEN ENGINEERING AND RESEARCH, INC.,14X,1H*/   TSFOIL  60
     2   24X,1H*,20X,26H MOUNTAIN VIEW, CALIFORNIA,21X,1H*)                TSFOIL  61
C                                                                          TSFOIL  62
C                 THE MAIN PROGRAM DOES NO COMPUTATIONS                    TSFOIL  63
C                 ALL COMPUTATIONS ARE DONE IN SUBROUTINES CALLED BY       TSFOIL  64
C                 TSFOIL.                                                  TSFOIL  65
C                 SUBROUTINE ECHINP PROVIDES A LISTING OF ALL DATA         TSFOIL  66
C                 CARDS FOR ENTIRE JOB.                                    TSFOIL  67
      CALL ECHINP                                                          TSFOIL  68
C                 SUBROUTINE READIN READS ALL INPUT AND CHECKS IT          TSFOIL  69
    1 CONTINUE                                                             TSFOIL  70
      CALL READIN                                                          TSFOIL  71
C                                                                          TSFOIL  72
C                 SUBROUTINE SCALE RESCALES ALL PHYSICAL VARIABLES TO      TSFOIL  73
C                 TRANSONIC SIMILARITY FORM                                TSFOIL  74
      CALL SCALE                                                           TSFOIL  75
C                                                                          TSFOIL  76
C                 SUBROUTINE FARFLD SETS FAR FIELD BOUNDARY CONDITIONS.    TSFOIL  77
      CALL FARFLD                                                          TSFOIL  78
C                                                                          TSFOIL  79
C                 SUBROUTINE BODY COMPUTES AIRFOIL GEOMETRY AND PRINTS     TSFOIL  80
C                 OUT GEOMETRICAL INFORMATION                              TSFOIL  81
      CALL BODY                                                            TSFOIL  82
C                                                                          TSFOIL  83
C                 SUBROUTINE CUTOUT REMOVES MESH POINTS FROM THE INPUT     TSFOIL  84
C                 MESH. CALCULATIONS ARE DONE FIRST ON COARSE MESH.        TSFOIL  85
C                 AND THEN ON PROGRESSIVELY REFINED MESHES UNTIL           TSFOIL  86
C                 INPUT MESH IS ACHIEVED.                                  TSFOIL  87
      CALL CUTOUT                                                          TSFOIL  88
C                                                                          TSFOIL  89
C                 SUBROUTINE GUESSP INITIALIZES P ARRAY                    TSFOIL  90
      CALL GUESSP                                                          TSFOIL  91
C                                                                          TSFOIL  92
C                 SUBROUTINE DIFCOE CALCULATES FINITE DIFFERENCE           TSFOIL  93
C                 EQUATION COEFFICIENTS WHICH DEPEND ON MESH SIZE.         TSFOIL  94
      CALL DIFCOE                                                          TSFOIL  95
C                                                                          TSFOIL  96
C                 SUBROUTINE SETBC  ADJUSTS THE AIRFOIL SLOPE BOUNDARY     TSFOIL  97
C                 CONDITION FOR THE CURRENT X Y MESH. ALSO, THE LIMITS     TSFOIL  98
C                 ON I AND J INDICIES FOR SOLVING THE DIFFERENCE           TSFOIL  99
C                 EQUATIONS ARE SET.                                       TSFOIL 100
      CALL SETBC                                                           TSFOIL 101
C                                                                          TSFOIL 102
C                 SUBROUTINE SOLVE EXECUTES THE MAIN RELAXATION            TSFOIL 103
C                 SOLUTION OF THE DIFFERENCE EQUATIONS                     TSFOIL 104
      CALL SOLVE                                                           TSFOIL 105
C                                                                          TSFOIL 106
C                 IF FINAL MESH HAS BEEN REACHED, RESULTS ARE PRINTED      TSFOIL 107
C                 OUT IN FINAL FORM.  IF NOT, INTERMEDIATE RESULTS         TSFOIL 108
C                 ARE PRINTED OUT AND THE MESH IS REFINED.  THE ABOVE      TSFOIL 109
C                 SEQUENCE OF CALCULATIONS ARE THEN REPEATED               TSFOIL 110
      IF(IREF .LE. 0) GO TO 5                                              TSFOIL 111
      IF (ABORT) GO TO 5                                                   TSFOIL 112
C                                                                          TSFOIL 113
C                 SUBROUTINE PRINT1 PRINTS OUT BODY PRESSURE               TSFOIL 114
C                 DISTRIBUTION.                                            TSFOIL 115
      CALL PRINT1                                                          TSFOIL 116
C                                                                          TSFOIL 117
C                 SUBROUTINE REFINE  ADDS MESH POINTS                      TSFOIL 118
      CALL REFINE                                                          TSFOIL 119
C                                                                          TSFOIL 120
C                 REPEAT SEQUENCE OF RELAXATION CALCULATIONS               TSFOIL 121
      CALL DIFCOE                                                          TSFOIL 122
      CALL SETBC                                                           TSFOIL 123
      CALL SOLVE                                                           TSFOIL 124
C                                                                          TSFOIL 125
C                 IF FINAL MESH HAS BEEN REACHED, RESULTS ARE PRINTED      TSFOIL 126
C                 OUT IN FINAL FORM.  IF NOT, INTERMEDIATE RESULTS         TSFOIL 127
C                 ARE PRINTED OUT AND THE MESH IS REFINED.  THE ABOVE      TSFOIL 128
C                 SEQUENCE OF CALCULATIONS ARE THEN REPEATED               TSFOIL 129
      IF(IREF .LE. 0) GO TO 5                                              TSFOIL 130
      IF (ABORT) GO TO 5                                                   TSFOIL 131
      CALL PRINT1                                                          TSFOIL 132
      CALL REFINE                                                          TSFOIL 133
      CALL DIFCOE                                                          TSFOIL 134
      CALL SETBC                                                           TSFOIL 135
      CALL SOLVE                                                           TSFOIL 136
C                                                                          TSFOIL 137
C                 RELAXATION SOLUTION IS COMPLETED                         TSFOIL 138
    5 CONTINUE                                                             TSFOIL 139
C                                                                          TSFOIL 140
C                 PRINTOUT FINAL INFORMATION                               TSFOIL 141
      CALL PRINT                                                           TSFOIL 142
      IF (IREF .GT. 0 ) CALL REFINE                                        TSFOIL 143
      IF (IREF .GT. 0 ) CALL REFINE                                        TSFOIL 144
C                                                                          TSFOIL 145
C                 STORE SOLUTION FOR NEXT CASE ON ON TAPE 3                TSFOIL 146
      CALL SAVEP                                                           TSFOIL 147
C                                                                          TSFOIL 148
C                 RETURN TO READIN TO COMPUTE NEXT CASE OR TERMINATE       TSFOIL 149
C                 CALCULATIONS.                                            TSFOIL 150
      GO TO 1                                                              TSFOIL 151
      END                                                                  TSFOIL 152
```

```
      SUBROUTINE ANGLE                                                     ANGLE    2
C                 COMPUTES THETA AT EVERY MESH POINT.                      ANGLE    3
C                 CALLED BY - FARFLD.                                      ANGLE    4
      COMMON / COM1/ IMIN     , IMAX     , IUP      , IDOWN    , ILE      , COM1     2
     1               ITE      , JMIN     , JMAX     , JUP      , JLOW     , COM1     3
     2               JTOP     , JBOT                                       COM1     4
      COMMON / COM2/ AK       , ALPHA    , DUB      , GAM1     , RTK       COM2     2
      LOGICAL        AMESH                                                 COM4     2
      COMMON / COM4/ XIN(100) , YIN(100), AMESH                            COM4     3
      COMMON /COM12/ F        , H        , HALFPI   , PI       , RTKPOR   , COM12    2
     1               TWOPI                                                 COM12    3
```

```
      REAL JET                                                          COM16     2
      COMMON /COM16/ ALPHAD   , ALPHA1  , ALPHA2  , XSING   ,           COM16     3
     1               OMEGAD   , OMEGA1  , OMEGA2  , JET                 COM16     4
      COMMON /COM33/ THETA(100,100)                                     COM33     2
                                                                        ANGLE    11
C                       SUBROUTINE TO COMPUTE THE ANGLE  THETA AT        ANGLE    12
C                       EACH MESH POINT.                                 ANGLE    13
                                                                        ANGLE    14
      R2PI = 1.0 / TWOPI                                                ANGLE    15
      DO 20 I=IMIN,IMAX                                                 ANGLE    16
      XX = XIN(I) - XSING                                               ANGLE    17
      DO 10 J=JMIN,JMAX                                                 ANGLE    18
      YY = YIN(J) + RTK                                                 ANGLE    19
      R = SQRT(YIN(J)**2 + XX*XX)                                       ANGLE    20
      ATN = ATAN2(YY,XX)                                                ANGLE    21
      D = PI - SIGN(PI,YY)                                              ANGLE    22
      THETA(J,I) = -(ATN + D) * R2PI                                    ANGLE    23
      IF ( R .LE. 1.0 ) THETA(J,I) = THETA(J,I) * R                     ANGLE    24
   10 CONTINUE                                                          ANGLE    25
   20 CONTINUE                                                          ANGLE    26
      RETURN                                                            ANGLE    27
      END                                                               ANGLE    28
```

```
      FUNCTION ARF (X)                                                  ARF       2
                                                                        ARF       3
C             EVALUATES ERF WITH AN ERROR .LT. 1.5E-7 BY RATIONAL        ARF       4
C             APPROXIMATION 7.1.26 OF HANDBOOK OF MATH. FUNCTIONS        ARF       5
C             U. S. DEPT. OF COMMERCE.  NBS APPL MATH SER 55.            ARF       6
C             CALLED BY - AYMESH.                                        ARF       7
                                                                        ARF       8
      DIMENSION C(5)                                                    ARF       9
      DATA C /1.061405429,-1.453152027,1.421413741, -.284496736,        ARF      10
     1        .254829592/                                               ARF      11
                                                                        ARF      12
      Y = X                                                             ARF      13
      IF ( X .LT. 0.0 ) Y = -Y                                          ARF      14
      IF ( Y .LT. 10. ) GO TO 10                                        ARF      15
      ARF = 1.0                                                         ARF      16
      GO TO 30                                                          ARF      17
   10 CONTINUE                                                          ARF      18
      T    = 1.0 / (1.0 + .3275911*Y)                                   ARF      19
      POLY = 0.0                                                        ARF      20
      DO 20 I=1,5                                                       ARF      21
      POLY = (POLY + C(I)) * T                                          ARF      22
   20 CONTINUE                                                          ARF      23
      ARF = 1.0 - POLY * EXP(-Y*Y)                                      ARF      24
   30 CONTINUE                                                          ARF      25
      IF (X .LT. 0.0) ARF = -ARF                                        ARF      26
      RETURN                                                            ARF      27
      END                                                               ARF      28
```

```
      SUBROUTINE AYMESH                                                 AYMESH    2
C                   COMPUTES ANALYTICAL X AND Y MESH POINTS.             AYMESH    3
C                   CALLED BY - READIN.                                  AYMESH    4
      COMMON / COM1/ IMIN     , IMAX    , IUP     , IDOWN   , ILE     ,  COM1      2
     1               ITE      , JMIN    , JMAX    , JUP     , JLOW    ,  COM1      3
     2               JTOP     , JBOT                                    COM1      4
      LOGICAL        AMESH                                              COM4      2
      COMMON / COM4/ XIN(100) , YIN(100), AMESH                         COM4      3
      INTEGER        PSTART                                             COM11     2
      LOGICAL        PSAVE                                              COM11     3
      COMMON /COM11/ ALPHAO   , CLOLD   , DELTAO  , DUBO    , EMACHO  , COM11     4
     1               IMINO    , IMAXO   , IMAXI   , JMINO   , JMAXO   , COM11     5
     2               JMAXI    , PSAVE   , PSTART  , TITLE(8), TITLEO(8), COM11    6
     3               VOLD     , XOLD(100),YOLD(100)                     COM11     7
      COMMON /COM12/ F        , H       , HALFPI  , PI      , RTKPOR  , COM12     2
     1               TWOPI                                              COM12     3
      INTEGER        BCTYPE                                             COM20     2
      COMMON /COM20/ BCTYPE   , CIRCFF  , FHINV   , POR     , CIRCTE    COM20     3
      COMMON /COM30/ BXH(401) , REST(3)                                 AYMESH   10
                                                                        AYMESH   11
      DIMENSION  XH(401) , BX(100)                                      AYMESH   12
      DATA  A0/.225/, A1/1.4/, A2/1.6/, A3/.6188/, A4/.75/, A5/30./,    AYMESH   13
     1      A6/.603/, A7/2.0/                                           AYMESH   14
      DATA CT1/2.0/, CT2/2.0/, CT3/1.0/, CF1/1.0/, CF2/1.0/, CF3/5.2/   AYMESH   15
                                                                        AYMESH   16
      IMAXI=81                                                          AYMESH   17
      JMAXI=64                                                          AYMESH   18
      IMA = (IMAXI - 1) / 2                                             AYMESH   19
      A22 = A2 * A2                                                     AYMESH   20
      A72 = A7 * A7                                                     AYMESH   21
      FACT = HALFPI * .005                                              AYMESH   22
      DO 10 I=201,401                                                   AYMESH   23
      EX     = TAN(FACT * (I-201) )                                     AYMESH   24
      XH(I)  = EX                                                       AYMESH   25
      EX2    = EX * EX                                                  AYMESH   26
      BXH(I) = A1 * EX * EXP(-A22*EX2) + (1.0-EXP(-A72*EX2))*ARF(A4*EX) AYMESH   27
   10 CONTINUE                                                          AYMESH   28
      DO 20 I=1,200                                                     AYMESH   29
      XH(I)  = - XH(402-I)                                              AYMESH   30
      BXH(I) = -BXH(402-I)                                              AYMESH   31
   20 CONTINUE                                                          AYMESH   32
      TOOPI = 2.0 / PI                                                  AYMESH   33
                                                                        AYMESH   34
C                    ADD 2/PI ATAN(A5(X+A6)) TO BX4 TO GIVE MORE         AYMESH   35
C                    POINTS NEAR THE LEADING EDGE.                       AYMESH   36
                                                                        AYMESH   37
      DO 30 I=1,401                                                     AYMESH   38
      BXH(I) = BXH(I) * (1.-A0) + TOOPI*A0*ATAN(A5*(XH(I)+A6))          AYMESH   39
   30 CONTINUE                                                          AYMESH   40
                                                                        AYMESH   41
      DX    = 1.0 / IMA                                                 AYMESH   42
      IM2   = IMA * 2                                                   AYMESH   43
      IM2P1= IM2 + 1                                                    AYMESH   44
                                                                        AYMESH   45
      DO 42 I=1,IM2P1                                                   AYMESH   46
      IF (I .EQ. 1 .OR. I .EQ. IM2P1) GO TO 31                          AYMESH   47
      BX(I) = (I-1) * DX - 1.0                                          AYMESH   48
      GO TO 32                                                          AYMESH   49
   31 CONTINUE                                                          AYMESH   50
      IF (I .EQ. 1) BX(I) = -.999                                       AYMESH   51
      IF (I .EQ. IM2P1) BX(I) = .999                                    AYMESH   52
   32 CONTINUE                                                          AYMESH   53
      J     = 0                                                         AYMESH   54
   33 CONTINUE                                                          AYMESH   55
      J     = J + 1                                                     AYMESH   56
      IF (BX(I) .GT. BXH(J)) GO TO 33                                   AYMESH   57
      IF (BX(I) .LT. BXH(J)) GO TO 34                                   AYMESH   58
      XI    = XH(J)                                                     AYMESH   59
      GO TO 40                                                          AYMESH   60
                                                                        AYMESH   61
   34 CONTINUE                                                          AYMESH   62
      BT1  = BX(I)  - BXH(J-1)                                          AYMESH   63
      BHT1 = BXH(J) - BXH(J-1)                                          AYMESH   64
      XHT1 = XH(J)  - XH(J-1)                                           AYMESH   65
      T1   = XHT1 / BHT1                                                AYMESH   66
      XI    = XH(J-1) + BT1 * T1                                        AYMESH   67
      IF (J .EQ. 2) GO TO 40                                            AYMESH   68
                                                                        AYMESH   69
      BT2  = BX(I)    - BXH(J)                                          AYMESH   70
      BHT2 = BXH(J)   - BXH(J-2)                                        AYMESH   71
      BHT3 = BXH(J-1) - BXH(J-2)                                        AYMESH   72
      XHT2 = XH(J-1)  - XH(J-2)                                         AYMESH   73
      T2   = XHT2 / BHT3                                                AYMESH   74
      T12  = T1 - T2                                                    AYMESH   75
      BT12 = BT1 * BT2                                                  AYMESH   76
      TBT2 = T12 / BHT2                                                 AYMESH   77
      XI    = XI + BT12 * TBT2                                          AYMESH   78
      IF (J .GE. 400) GO TO 40                                          AYMESH   79
                                                                        AYMESH   80
      BT3  = BX(I)    - BXH(J-2)                                        AYMESH   81
      BHT4 = BXH(J+1) - BXH(J-2)                                        AYMESH   82
      BHT5 = BXH(J+1) - BXH(J)                                          AYMESH   83
      BHT6 = BXH(J+1) - BXH(J-1)                                        AYMESH   84
      XHT3 = XH(J+1)  - XH(J)                                           AYMESH   85
      T3   = XHT3 / BHT5                                                AYMESH   86
      XI    = XI + BT12 * BT3/BHT4 * ((T3-T1)/BHT6 - TBT2)              AYMESH   87
   40 CONTINUE                                                          AYMESH   88
      XIN(I) = XI + A3                                                  AYMESH   89
   42 CONTINUE                                                          AYMESH   90
      CALL ISLIT(XIN)                                                   AYMESH   91
      XFACT=1.0/XIN(ITE)                                                AYMESH   92
      DO 43 I=1,IMAXI                                                   AYMESH   93
      XIN(I)=XIN(I)*XFACT                                               AYMESH   94
   43 CONTINUE                                                          AYMESH   95
      XIN(ITE)=1.0                                                      AYMESH   96
      IMAXI = IM2P1                                                     AYMESH   97
      JM2   = JMAXI                                                     AYMESH   98
```

APPENDIX

```
      JMA = JM2 / 2                                                     AYMESH  99
      FJ = JMA                                                          AYMESH 100
      IF (BCTYPE .NE. 1) GO TO 44                                       AYMESH 101
      C1    = CF1                                                       AYMESH 102
      C2    = CF2                                                       AYMESH 103
      C3    = CF3                                                       AYMESH 104
      GO TO 45                                                          AYMESH 105
   44 CONTINUE                                                          AYMESH 106
      C1    = CT1                                                       AYMESH 107
      C2    = CT2                                                       AYMESH 108
      C3    = CT3                                                       AYMESH 109
   45 CONTINUE                                                          AYMESH 110
      DETA = 1.0 / (FJ*C1)                                              AYMESH 111
      IF (BCTYPE .EQ. 1) DETA = 1.0 / ((FJ+1.0)*C1)                     AYMESH 112
      C     = C3 / (TAN(HALFPI*DETA*FJ))**C2                            AYMESH 113
      DO 50 I=1,JMA                                                     AYMESH 114
      J     = JMA + I                                                   AYMESH 115
      ETAI = I * DETA                                                   AYMESH 116
      YIN(J) = C * (TAN(HALFPI*ETAI))**C2                               AYMESH 117
      YIN(J-2*I+1)   = -YIN(J)                                          AYMESH 118
   50 CONTINUE                                                          AYMESH 119
      RETURN                                                            AYMESH 120
      END                                                               AYMESH 121
```

```
      BLOCK DATA                                                        BLKDT    2
                                                                        BLKDT    3
                                                                        BLKDT    4
      COMMON          P(107,101),X(130)  , Y(100)                       BLANK    2
      COMMON / COM1/ IMIN     , IMAX    , IUP     , IDOWN   , ILE     , COM1     2
     1               ITE      , JMIN    , JMAX    , JUP     , JLOW    , COM1     3
     2               JTOP     , JBOT                                    COM1     4
      COMMON / COM2/ AK       , ALPHA   , DUB     , GAM1    , RTK       COM2     2
      LOGICAL        ABORT                                              COM3     2
      COMMON / COM3/ IREF     , ABORT   , ICUT    , KSTEP               COM3     3
      LOGICAL        AMESH                                              COM4     2
      COMMON / COM4/ XIN(100) , YIN(100), AMESH                         COM4     3
      COMMON / COM5/ XDIFF(100),YDIFF(100)                              COM5     2
      COMMON / COM6/ FL(100)  , FXL(100), FU(100) , FXU(100),           COM6     2
     1           CAMBER(100), THICK(100),VOL    , XFOIL(100), IFOIL     COM6     3
      COMMON / COM7/ CJUP     , CJUP1   , CJLOW   , CJLOW1              COM7     2
      COMMON / COM8/ CVERGE   , DVERGE  , IPRTER  , MAXIT   ,           COM8     2
     1               WE(3)    , EPS                                     COM8     3
      INTEGER        BCFOIL                                             COM9     2
      COMMON / COM9/ BCFOIL   , NL      , NU      , XL(100) , XU(100) , COM9     3
     1               YL(100)  , YU(100) , RIGF                          COM9     4
      LOGICAL XGRDIN,YGRDIN                                             COM10    2
      COMMON /COM10/ YFREE(100) , YTUN(100)  , XKRUPP(100) , GAM      , COM10    3
     1               JMXF     , JMXT    , XGRDIN  , YGRDIN              COM10    4
      INTEGER        PSTART                                             COM11    2
      LOGICAL        PSAVE                                              COM11    3
      COMMON /COM11/ ALPHAO   , CLOLD   , DELTAO  , DUBO    , EMACHO  , COM11    4
     1               IMINO    , IMAXO   , IMAXI   , JMINO   , JMAXO   , COM11    5
     2               JMAXI    , PSAVE   , PSTART  , TITLE(8), TITLEO(8),COM11    6
     3               VOLO     , XOLD(100),YOLD(100)                     COM11    7
      COMMON /COM12/ F        , H       , HALFPI  , PI      , RTKPOR  , COM12    2
     1               TWOPI                                              COM12    3
      COMMON /COM13/ CDFACT   , CLFACT  , CMFACT  , CPFACT  , CPSTAR    COM13    2
      LOGICAL        FCR      , KUTTA                                   COM14    2
      COMMON /COM14/ CLSET    , FCR     , KUTTA   , WCIRC               COM14    3
      COMMON /COM15/ B        , BETA0   , BETA1   , BETA2   , PSI0    , COM15    2
     1               PSI1     , PSI2                                    COM15    3
      REAL JET                                                          COM16    2
      COMMON /COM16/ ALPHA0   , ALPHA1  , ALPHA2  , XSING   ,           COM16    3
     1               OMEGA0   , OMEGA1  , OMEGA2  , JET                 COM16    4
      COMMON /COM17/ CYYBLC   , CYYBLD  , CYYBLU  , CYYBUC  , CYYBUD  , COM17    2
     1               CYYBUU   ,FXLBC(100),FXUBC(100)                    COM17    3
      LOGICAL        OUTERR                                             COM18    2
      COMMON /COM18/ ERROR    , I1      , I2      , IERROR  , JERROR  , COM18    3
     1               OUTERR   , EMU(100,2)        , VC(100) ,           COM18    4
     2               WI       , DCIRC   , POLD(100,2)                   COM18    5
      COMMON /COM19/ DIAG(100), RHS(100), SUB(100), SUP(100)            COM19    2
      COMMON /COM20/ AMID(100), YMID(100)                               COM20    2
      COMMON /COM22/ CXC(100) , CXL(100), CXR(100), CXXC(100),CXXL(100),COM22    2
     1               CXXR(100), C1(100)                                 COM22    3
      COMMON /COM23/ CYYC(100), CYYD(100),CYYU(100), IVAL               COM23    2
      COMMON /COM24/ DTOP(100), DBOT(100),DUP(100), DDOWN(100),         COM24    2
     1               VTOP(100), VBOT(100),VUP(100), VDOWN(100)          COM24    3
      COMMON /COM25/ CPL(100) , CPU(100)                                COM25    2
```

```
      COMMON /COM26/ PJUMP(100)                                         COM26    2
      LOGICAL        PHYS                                               COM27    2
      INTEGER        PRTFLO   , SIMDEF                                  COM27    3
      COMMON /COM27/ CL       , DELTA   , DELRT2  , EMACH   , EMROOT  , COM27    4
     1               PHYS     , PRTFLO  , SIMDEF  , SONVEL  , VFACT   , COM27    5
     2               YFACT                                              COM27    6
      INTEGER        BCTYPE                                             COM28    2
      COMMON /COM28/ BCTYPE   , CIRCFF  , FHINV   , POR     , CIRCTE    COM28    3
      COMMON /COM32/ BIGRL    , IRL     , JRL                           COM32    2
      COMMON /COM33/ THETA(100,100)                                     COM33    2
                                                                        BLKDT   35
      DATA     BCFOIL / 3 / , BCTYPE / 1 /  , PSTART / 1 / ,            BLKDT   36
     1         PRTFLO / 1 / , SIMDEF / 3 /                              BLKDT   37
      DATA     PHYS /.T. /  , PSAVE / .F. / , FCR  / .T. / ,            BLKDT   38
     1         KUTTA / .T. /, ABORT / .T. /,  AMESH / .F. /             BLKDT   39
      DATA     EMACH  /.75/ , DELTA /.115/ , ALPHA   /.12/ ,            BLKDT   40
     1         AK     /0.0/,  GAM/1.4/, RIGF/0.0/, EPS/.2/              BLKDT   41
      DATA     CLSET /0.0/,  CVERGE/.00001/,  DVERGE/10./,              BLKDT   42
     1         F      / 0.0 /, H     / 0.0 /  , POR     / 0.0/,         BLKDT   43
     1         WCIRC/1.0/,  WE/1.8,1.9,1.95/                            BLKDT   44
      DATA     XGRDIN /.F./, YGRDIN /.F./                               BLKDT   45
      DATA     IMAXI/77/,  JMXF/56/,  MAXIT/500/,                       BLKDT   46
     1         NL     / 75  /, NU    / 100 /  , IPRTER / 10 /           BLKDT   47
      DATA     PI /3.14159265/ , HALFPI /1.570796325/ ,                 BLKDT   48
     1         TWOPI/ 6.28318531/                                       BLKDT   49
      DATA    IMIN /1/       , JMIN /1/       , ICUT /2/                BLKDT   50
      DATA CPU/100*0.0/,CPL/100*0.0/                                    BLKDT   51
      DATA   YIN /100*0.0  /, JMAXI /100 /  , JMXT    /48 /             BLKDT   52
      DATA XKRUPP / -1.075    , -.955    ,                              BLKDT   53
     1            -.825    , -.7      , -.575    , -.45     , -.35     , BLKDT   54
     2            -.25     , -.175    , -.125    , -.075    , -.0525   , BLKDT   55
     3            -.035    , -.0225   , -.015    , -.0075   , -.0025   , BLKDT   56
     4             .0025   ,  .0075   ,  .0125   ,  .0175   ,  .0225   , BLKDT   57
     5             .0275   ,  .0325   ,  .0375   ,  .045    ,  .055    , BLKDT   58
     6             .065    ,  .075    ,  .085    ,  .0975   ,  .115    , BLKDT   59
     7             .140625 ,  .171875,  .203125,  .234375,  .265625,   BLKDT   60
     8             .296875 ,  .328125,  .359375,  .390625,  .421875,   BLKDT   61
     9             .453125 ,  .484375,  .515625,  .546875,  .578125,   BLKDT   62
     1             .609375 ,  .640625,  .671875,  .703125,  .734375,   BLKDT   63
     2             .765625 ,  .796875,  .828125,  .859375,  .885    ,  BLKDT   64
     3             .9      ,  .915    ,  .93     ,  .945    ,  .96     , BLKDT   65
     4             .975    ,  .99     ,  1.0     ,  1.01    ,  1.025   , BLKDT   66
     5             1.05    ,  1.09    ,  1.15    ,  1.225   ,  1.3     , BLKDT   67
     6             1.4     ,  1.5     ,  1.625   ,  1.75    ,  1.875   , BLKDT   68
     7             25*0.0 /                                             BLKDT   69
      DATA YFREE /  -5.2    , -4.4    , -3.6    , -3.0    , -2.4    ,   BLKDT   70
     1              -1.95   , -1.6    , -1.35   , -1.15   , -.95    ,   BLKDT   71
     2              -.80    , -.65    , -.55    , -.45    , -.39    ,   BLKDT   72
     3              -.34    , -.30    , -.27    , -.24    , -.21    ,   BLKDT   73
     4              -.18    , -.15    , -.125   , -.1     , -.075   ,   BLKDT   74
     5              - .05   , - .03   , - .01   ,    .01  ,    .03  ,   BLKDT   75
     6                 .05  ,   .075  ,    .1   ,   .125  ,    .15  ,   BLKDT   76
     7                 .18  ,    .21  ,    .24  ,    .27  ,    .30  ,   BLKDT   77
     8                 .34  ,    .39  ,    .45  ,    .55  ,    .65  ,   BLKDT   78
     9                 .8   ,    .95  ,   1.15  ,   1.35  ,   1.60  ,   BLKDT   79
     1               1.95   ,   2.4   ,   3.0   ,   3.6   ,   4.4   ,   BLKDT   80
     2               5.2    ,  44*0.0   /                               BLKDT   81
      DATA   YTUN /  -2.0   ,  -1.8   ,  -1.6   ,  -1.4   , -1.2    ,   BLKDT   82
     1              -1.0    ,   -.8   ,   -.65  ,   -.55  , -.45    ,   BLKDT   83
     2               -.39   ,   -.34  ,   -.30  ,   -.27  , -.24    ,   BLKDT   84
     3               -.21   ,   -.18  ,   -.15  ,  -.125  , -.1     ,   BLKDT   85
     4              - .075  ,  - .05  ,  - .03  ,  - .01  ,    .01  ,   BLKDT   86
     5                .03   ,    .05  ,    .075 ,    .1   ,   .125  ,   BLKDT   87
     6                .15   ,    .18  ,    .21  ,    .24  ,   .27   ,   BLKDT   88
     7                .3    ,    .34  ,    .39  ,    .45  ,   .55   ,   BLKDT   89
     8                .65   ,    .8   ,   1.0   ,   1.2   ,   1.4   ,   BLKDT   90
     9               1.6    ,   1.8   ,   2.0   ,  52*0.0  /            BLKDT   91
      DATA   XU   /  0.000008, 0.000167, 0.000391, 0.000709, 0.001407,  BLKDT   92
     1               0.002153, 0.003331, 0.005336, 0.008848, 0.014303,  BLKDT   93
     2               0.023401, 0.033091, 0.040007, 0.053973, 0.056921,  BLKDT   94
     3               0.058456, 0.059966, 0.061445, 0.062905, 0.063925,  BLKDT   95
     4               0.068705, 0.071482, 0.074007, 0.075822, 0.076608,  BLKDT   96
     5               0.077862, 0.079112, 0.080445, 0.081819, 0.083209,  BLKDT   97
     6               0.084841, 0.086702, 0.088848, 0.091378, 0.094413,  BLKDT   98
     7               0.098308, 0.103104, 0.109010, 0.116244, 0.125452,  BLKDT   99
     8               0.136635, 0.150037, 0.165853, 0.184699, 0.195177,  BLKDT  100
     9               0.206361, 0.218244, 0.230813, 0.244047, 0.257917,  BLKDT  101
     1               0.272371, 0.287410, 0.302990, 0.319097, 0.335595,  BLKDT  102
     2               0.352421, 0.369591, 0.386995, 0.404133, 0.421391,  BLKDT  103
     3               0.438706, 0.456013, 0.473246, 0.490341, 0.507242,  BLKDT  104
     4               0.523881, 0.539536, 0.554867, 0.569823, 0.584351,  BLKDT  105
     5               0.598405, 0.611936, 0.624904, 0.637273, 0.648483,  BLKDT  106
     6               0.659016, 0.668987, 0.678321, 0.687012, 0.693090,  BLKDT  107
     7               0.706936, 0.720404, 0.730649, 0.761390, 0.777016,  BLKDT  108
     8               0.792241, 0.809060, 0.824992, 0.838953, 0.857100,  BLKDT  109
     9            0.875621, 0.890208, 0.913606, 0.927606, 0.939804,      BLKDT 110
     1            0.952002, 0.971766, 0.989100, 0.997746, 1.000000/      BLKDT 111
      DATA YU /   0.000787, 0.003092, 0.004538, 0.006137, 0.007603,      BLKDT 112
     1            0.009056, 0.010675, 0.012801, 0.013607, 0.019624,      BLKDT 113
     2            0.024441, 0.029325, 0.031698, 0.033946, 0.036037,      BLKDT 114
     3            0.037277, 0.037706, 0.038103, 0.038497, 0.039276,      BLKDT 115
     4            0.039986, 0.040625, 0.041195, 0.041481, 0.041736,      BLKDT 116
     5            0.042019, 0.042274, 0.042539, 0.042804, 0.043076,      BLKDT 117
     6            0.043368, 0.043700, 0.044072, 0.044407, 0.044805,      BLKDT 118
     7            0.045345, 0.046112, 0.047154, 0.048132, 0.049381,      BLKDT 119
     8            0.050628, 0.052209, 0.053663, 0.055351, 0.056210,      BLKDT 120
     9            0.057068, 0.057914, 0.058731, 0.059559, 0.060335,      BLKDT 121
     1            0.061018, 0.061751, 0.062381, 0.062947, 0.063445,      BLKDT 122
     2            0.063867, 0.064213, 0.064473, 0.064646, 0.064735,      BLKDT 123
     3            0.064733, 0.064651, 0.064473, 0.064218, 0.063871,      BLKDT 124
     4            0.063438, 0.062945, 0.062376, 0.061731, 0.061014,      BLKDT 125
     5            0.060232, 0.059384, 0.058496, 0.057562, 0.056650,      BLKDT 126
     6            0.055721, 0.054761, 0.053867, 0.052963, 0.052086,      BLKDT 127
     7            0.050722, 0.048045, 0.046800, 0.043441, 0.041051,      BLKDT 128
     8            0.038464, 0.035780, 0.032958, 0.030775, 0.026934,      BLKDT 129
     9            0.024361, 0.021046, 0.015756, 0.012954, 0.010567,      BLKDT 130
     1            0.008213, 0.004339, 0.001620, 0.000293, 0.000000/      BLKDT 131
      DATA XL /   0.000000, 0.000012, 0.000043, 0.000103, 0.000244,      BLKDT 132
     1            0.000348, 0.000455, 0.000686, 0.001011, 0.001401,      BLKDT 133
     2            0.001875, 0.002816, 0.003055, 0.004281, 0.004747,      BLKDT 134
     3            0.005774, 0.007035, 0.008765, 0.009969, 0.011220,      BLKDT 135
     4            0.013346, 0.016276, 0.025335, 0.029376, 0.035005,      BLKDT 136
     5            0.052516, 0.062469, 0.073324, 0.085290, 0.098822,      BLKDT 137
     6            0.118563, 0.140987, 0.167184, 0.228936, 0.228511,      BLKDT 138
     7            0.247655, 0.263995, 0.282047, 0.297845, 0.310147,      BLKDT 139
     8            0.324075, 0.344072, 0.363562, 0.387844, 0.404402,      BLKDT 140
     9            0.420968, 0.450216, 0.475376, 0.521037, 0.549043,      BLKDT 141
     1            0.578612, 0.603381, 0.623478, 0.642152, 0.657943,      BLKDT 142
     2            0.674212, 0.690340, 0.708001, 0.726806, 0.744603,      BLKDT 143
     3            0.766102, 0.784092, 0.801144, 0.819167, 0.838344,      BLKDT 144
     4            0.858817, 0.879431, 0.903723, 0.926504, 0.943652,      BLKDT 145
     5            0.956669, 0.973623, 0.986187, 0.996582, 1.000000,      BLKDT 146
     6            25*0.0/                                                BLKDT 147
      DATA YL /   0.000000,-0.000767,-0.001505,-0.002066,-0.003330,      BLKDT 148
     1           -0.003866,-0.004370,-0.005199,-0.006131,-0.007103,      BLKDT 149
     2           -0.007933,-0.008676,-0.009776,-0.011204,-0.011015,      BLKDT 150
     3           -0.012861,-0.013902,-0.014962,-0.016175,-0.017606,      BLKDT 151
     4           -0.019136,-0.021234,-0.023036,-0.025375,-0.028024,      BLKDT 152
     5           -0.032423,-0.034040,-0.037102,-0.039456,-0.041062,      BLKDT 153
     6           -0.044463,-0.047017,-0.049296,-0.051443,-0.052406,      BLKDT 154
     7           -0.052099,-0.053062,-0.053117,-0.053027,-0.052066,      BLKDT 155
     8           -0.052962,-0.051951,-0.051216,-0.050013,-0.049906,      BLKDT 156
     9           -0.047493,-0.045601,-0.043200,-0.038330,-0.034916,      BLKDT 157
     1           -0.031104,-0.027333,-0.024661,-0.021894,-0.019517,      BLKDT 158
     2           -0.017429,-0.014327,-0.011771,-0.008224,-0.006937,      BLKDT 159
     3           -0.003000,-0.002000,-0.000524, 0.000950, 0.002227,      BLKDT 160
     4            0.003224, 0.003865, 0.004212, 0.004067, 0.003657,      BLKDT 161
     5            0.003067, 0.002242, 0.001329, 0.000376, 0.000000,      BLKDT 162
     6            25*0.0/                                                BLKDT 163
      END                                                                BLKDT 164
```

```
      SUBROUTINE BCEND                                                   BCEND   2
C               SUBROUTINE BCEND MODIFIES THE DIAG AND RHS VECTORS       BCEND   3
C               ON EACH I LINE IN THE APPROPRIATE WAY TO INCLUDE THE     BCEND   4
C               BOUNDARY CONDITIONS AT JBOT AND JTOP.                    BCEND   5
C               CALLED BY - SYOR.                                        BCEND   6
                                                                         BCEND   7
      COMMON          P(102,101),X(100)  , Y(100)                        BLANK   2
      COMMON / COM1/  IMIN     , IMAX    , IUP     , IDOWN   , ILE    ,  COM1    2
     1                ITE      , JMIN    , JMAX    , JUP     , JLOW   ,  COM1    3
     2                JTOP     , JBOT                                    COM1    4
      COMMON / COM2/  AK       , ALPHA   , DUB     , GAM1    , RTK       COM2    2
      COMMON / COM5/  XDIFF(100),YDIFF(100)                              COM5    2
      COMMON /COM19/  DIAG(100), RHS(100), SUB(100), SUP(100)            COM19   2
      COMMON /COM23/  CYYC(100), CYYD(100),CYYU(100), IVAL               COM23   2
      INTEGER         BCTYPE                                             COM26   2
      COMMON /COM28/  BCTYPE   , CIRCFF  , PHINV   , POR     , CIRCTE    COM28   3
                                                                         BCEND  15
      I = IVAL                                                           BCEND  16
C               BRANCH TO APPROPRIATE ADDRESS FOR BCTYPE                 BCEND  17
      GO TO (10,20,30,40,50,60) , BCTYPE                                 BCEND  18
C               BCTYPE = 1, FREE AIR                                     BCEND  19
 10   CONTINUE                                                           BCEND  20
C                                                                        BCEND  21
C               DIRICHLET BOUNDARY CONDITION FOR SUBSONIC FREESTREAM     BCEND  22
      IF (AK .GT. 0.0) RETURN                                            BCEND  23
C               NEUMAN BOUNDARY CONDITION FOR SUPERSONIC FREESTREAM      BCEND  24
      DFACL = -CYYD(JBOT) * RTK * XDIFF(I)                               BCEND  25
      DFACU = -CYYU(JTOP) * RTK * XDIFF(I)                               BCEND  26
      RFACL = DFACL * (P(JMIN,I) - P(JMIN,I-1))                          BCEND  27
      RFACU = DFACU * (P(JMAX,I) - P(JMAX,I-1))                          BCEND  28
      GO TO 95                                                           BCEND  29
C               BCTYPE = 2, SOLID WALL                                   BCEND  30
 20   CONTINUE                                                           BCEND  31
C               NEUMAN BOUNDARY CONDITION = 0.                           BCEND  32
C               NO MODIFICATION NECESSARY TO DIAG OR RHS                 BCEND  33
      DFACL=0.                                                           BCEND  34
      DFACU=0.                                                           BCEND  35
      RFACL=0.                                                           BCEND  36
      RFACU=0.                                                           BCEND  37
      GO TO 95                                                           BCEND  38
C               BCTYPE = 3, FREE JET                                     BCEND  39
 30   CONTINUE                                                           BCEND  40
C               DIRICHLET BOUNDARY CONDITION                             BCEND  41
      IF (AK .LT. 0.0) GO TO 31                                          BCEND  42
      PJMIN = -.75 * CIRCFF                                              BCEND  43
      PJMAX = -.25 * CIRCFF                                              BCEND  44
      GO TO 32                                                           BCEND  45
   31 CONTINUE                                                           BCEND  46
      PJMIN = 0.0                                                        BCEND  47
      PJMAX = 0.0                                                        BCEND  48
   32 CONTINUE                                                           BCEND  49
      GO TO 90                                                           BCEND  50
C               BCTYPE = 4, IDEAL SLOTTED WALL                           BCEND  51
 40   CONTINUE                                                           BCEND  52
C               NEUMAN BOUNDARY CONDITION                                BCEND  53
      DFACL =-PHINV * CYYD(JBOT)                                         BCEND  54
      DFACU =-PHINV * CYYU(JTOP)                                         BCEND  55
      IF (AK .LT. 0.0) GO TO 41                                          BCEND  56
      RFACL =  DFACL * ( .75 * CIRCFF + P(JBOT,I))                       BCEND  57
      RFACU =  DFACU * ( .25 * CIRCFF + P(JTOP,I))                       BCEND  58
      GO TO 42                                                           BCEND  59
   41 CONTINUE                                                           BCEND  60
      RFACL = DFACL * P(JBOT,I)                                          BCEND  61
      RFACU = DFACU * P(JTOP,I)                                          BCEND  62
   42 CONTINUE                                                           BCEND  63
      GO TO 95                                                           BCEND  64
C               BCTYPE = 5,  POROUS/PERFORATED WALL                      BCEND  65
 50   CONTINUE                                                           BCEND  66
      IF(POR .GT. 1.5) GO TO 55                                          BCEND  67
C               NEUMAN BOUNDARY CONDITION FOR POR .LT. 1.5               BCEND  68
      DFACL = -CYYD(JBOT) * POR * XDIFF(I)                               BCEND  69
      DFACU = -CYYU(JTOP) * POR * XDIFF(I)                               BCEND  70
      RFACL = DFACL * (P(JMIN,I) - P(JMIN,I-1))                          BCEND  71
      RFACU = DFACU * (P(JMAX,I) - P(JMAX,I-1))                          BCEND  72
      GO TO 95                                                           BCEND  73
 55   CONTINUE                                                           BCEND  74
C               DIRCHLET BOUNDARY CONDITION FOR POR .GT. 1.5             BCEND  75
      IF (I .NE. IUP) RETURN                                             BCEND  76
C               SET VALUES OF P ON BOUNDARY BY INTEGRATING PX USING      BCEND  77
C               OLD VALUES OF POTENTIAL                                  BCEND  78
      PJMIN = P(JMIN,IUP)                                                BCEND  79
      TERM = -.5 / (POR * (Y(JMIN) - Y(JMIN-1)))                         BCEND  80
      DO 57 II=IUP,IDOWN                                                 BCEND  81
      P(JMIN,II) = P(JMIN,II-1) - TERM * (X(II)-X(II-1)) *               BCEND  82
     1           (P(JMIN,II)+P(JMIN,II-1)-P(JMIN+1,II)-P(JMIN+1,II-1))   BCEND  83
   57 CONTINUE                                                           BCEND  84
      PJMAX = P(JMAX,IUP)                                                BCEND  85
      TERM = .5 / (POR * (Y(JMAX) - Y(JMAX-1)))                          BCEND  86
      DO 58 II=IUP,IDOWN                                                 BCEND  87
      P(JMAX,II) = P(JMAX,II-1) - TERM*(X(II) - X(II-1)) *               BCEND  88
     1           (P(JMAX,II)+P(JMAX,II-1)-P(JMAX-1,II)-P(JMAX-1,II-1))   BCEND  89
   58 CONTINUE                                                           BCEND  90
      RHS(JBOT) = RHS(JBOT) -(CYYD(JBOT)*(P(JBOT-1,I)-PJMIN))            BCEND  91
      RHS(JTOP) = RHS(JTOP) - (CYYU(JTOP)*(P(JTOP+1,I)-PJMAX))           BCEND  92
      RETURN                                                             BCEND  93
C               BCTYPE = 6,  GENERAL WALL BOUNDARY CONDITION             BCEND  94
C               DIFFERENCE EQUATIONS FOR THIS BOUNDARY CONDITION         BCEND  95
C               HAVE NOT YET BEEN WORKED OUT.  USER MUST INSERT          BCEND  96
C               INFORMATION NEEDED FOR CALCULATION                       BCEND  97
 60   CONTINUE                                                           BCEND  98
      WRITE(6,1000)                                                      BCEND  99
 1000 FORMAT(34H1ABNORMAL STOP IN SUBROUTINE BCEND/                      BCEND 100
     1       24H BCTYPE=6 IS NOT USEABLE)                                BCEND 101
      STOP                                                               BCEND 102
C                                                                        BCEND 103
C               DIRICHLET BOUNDARY CONDITIONS                            BCEND 104
 90   CONTINUE                                                           BCEND 105
```

```
      RHS(I,JBOT) = RHS(I,JBOT) - (CYYD(JBOT)*(P(JMIN,I)-P(JBOT+1,I)))
      RHS(I,JTOP) = RHS(I,JTOP) - (CYYU(JTOP)*(P(JMAX,I)-P(JTOP-1,I)))
      RETURN
C
C                   NEUMAN BOUNDARY CONDITIONS
   40 CONTINUE
      DIAG(I,JBOT) = DIAG(I,JBOT) + DFACL
      DIAG(I,JTOP) = DIAG(I,JTOP) + DFACU
      RHS(I,JBOT) = RHS(I,JBOT) - RFACL + CYYD(JBOT)*P(I,JBOT+1,I)
      RHS(I,JTOP) = RHS(I,JTOP) - RFACU + CYYU(JTOP)*P(I,JTOP-1,I)
      RETURN
      END
```

```
      SUBROUTINE BODY
C              COMPUTES BODY GEOMETRY INFORMATION FOR BOUNDARY
C              CONDITIONS AND OUTPUT INFORMATION.  FOUR CHOICES OF
C              BODY DESCRIPTION ARE AVAILABLE AS FOLLOWS
C                   BCFOIL = 1  NACA BODY AIRFOIL
C                   BCFOIL = 2  PARABOLIC ARC AIRFOIL
C                   BCFOIL = 3  AIRFOIL ORDINATES READ IN
C                   BCFOIL = 4  EXTRA ADDRESS FOR USERS CHOICE
C              BODY ORDINATES AND SLOPES ARE COMPUTED AT THE INPUT
C              X MESH LOCATIONS AND ARE DIVIDED BY THE THICKNESS
C              RATIO DELTA.  THE BODY VOLUME, CAMBER, AND THICKNESS
C              ARE ALSO COMPUTED.
C              ACTUAL BOUNDARY CONDITION IS SET IN SUBROUTINE SETBC
C              CALLED BY - TSFOIL.

      COMMON / COM1/ IMIN     , IMAX    , IUP     , IDOWN   , ILE     ,
     1               ITE      , JMIN    , JMAX    , JUP     , JLOW    ,
     2               JTOP     , JBOT
      COMMON / COM2/ AK       , ALPHA   , DUB     , GAM1    , RTK
      LOGICAL        AMESH
      COMMON / COM4/ XIN(100) , YIN(100), AMESH
      COMMON / COM6/ FL(100)  , FXL(100), FU(100) , FXU(100),
     1               CAMBER(100), THICK(100),VOL  , XFOIL(100), IFOIL
      INTEGER        BCFOIL
      COMMON / COM9/ BCFOIL   , NL      , NU      , XL(100) , XU(100) ,
     1               YL(100)  , YU(100) , RIGF
      LOGICAL        PHYS
      INTEGER        PRTFLO   , SIMDEF
      COMMON /COM27/ CL       , DELTA   , DELRT2  , EMACH   , EMROOT  ,
     1               PHYS     , PRTFLO  , SIMDEF  , SONVEL  , VFACT   ,
     2               YFACT
      COMMON /SPLN/  A(200)   , B(200)  , DY1     , DY2     , K1      ,
     1               K2       , XP      , YP      ,DYP

C              SET NUMBER OF POINTS ON AIRFOIL
      IFOIL = ITE - ILE + 1
C              ZERO ALL THICKNESSES AND SLOPES
      DO 10 I=IMIN,IMAX
      FU(I) = 0.
      FL(I) = 0.
      FXU(I) = 0.
      FXL(I) = 0.
   10 CONTINUE
C              BRANCH TO APPROPRIATE AIRFOIL SPECIFICATION
      GO TO (100,200,300,400), BCFOIL
  100 CONTINUE
C              BCFOIL = 1
C              FORMULA FOR NACA BODY SHAPE
      IC = 0
      DO 125 I = ILE,ITE
      IC = IC + 1
      Z = XIN(I)
      XFOIL(IC) = Z
      RTZ = SQRT(Z)
      Z2 = Z*Z
      Z3 = Z*Z2
      Z4 = Z*Z3
      FU(IC) = 1.4845*RTZ - .63*Z - 1.758*Z2 + 1.4215*Z3 - .5075*Z4
      FL(IC) = -FU(IC)
      FXU(IC) = .74225/RTZ - .63 - 3.516*Z + 4.2645*Z2 - 2.03*Z3
      FXL(IC) = -FXU(IC)
  125 CONTINUE
      GO TO 500
  200 CONTINUE
C              BCFOIL = 2
C              PARABOLIC ARC AIRFOIL*******BI-CONVEX.
      IC = 0
      DO 225 I=ILE,ITE
      IC = IC + 1
      Z = XIN(I)
      XFOIL(IC) = Z
      Z2 = Z*Z
      FU(IC) = 2.*(Z - Z2)
      FL(IC) =  -FU(IC)
      FXU(IC) = 2. - 4.*Z
      FXL(IC) = -FXU(IC)
  225 CONTINUE
      GO TO 500
  300 CONTINUE
C              BCFOIL = 3
C              BODY ORDINATES READ IN NAMELIST
      DELINV = 1.
      IF(PHYS) DELINV = 1./DELTA
C              COMPUTE ORDINATES AND SLOPES AT X MESH LOCATION ON
C              AIRFOIL BY CUBIC SPLINE INTERPOLATION.
C              DERIVATIVE END CONDITIONS ARE SPECIFIED AT X=0
C              AND X=1
      K1 = 1
      K2 = 1
C                          UPPER SURFACE
C              CALCULATE DY/DX AT END POINTS BY FINITE DIFFERENCE
C              FORMULA.
      DY1 = (YU(2) - YU(1)) / (XU(2) - XU(1))
      DY2 = (YU(NU) - YU(NU-1)) / (XU(NU) - XU(NU-1))
C              INITIALIZE CUBIC SPLINE INTERPOLATION
      CALL SPLN1(XU,YU,NU)
C              CALCULATE ORDINATES AND SLOPES
      IC = 0
      DO 320 I=ILE,ITE
      IC = IC + 1
      XP = XIN(I)
      XFOIL(IC) = XP
      CALL SPLN1X(XU,YU,NU)
C              SET ORDINATE AND SLOPE OF AIRFOIL TO INTERPOLATED
C              VALUE DIVIDED BY THICKNESS RATIO
      FU(IC) = YP*DELINV
      FXU(IC) = DYP*DELINV
  320 CONTINUE
C                          LOWER SURFACE
C              CALCULATE DY/DX AT END POINTS BY FINITE DIFFERENCE
C              FORMULA.
      DY1 = (YL(2) - YL(1)) / (XL(2) - XL(1))
      DY2 = (YL(NL) - YL(NL-1)) / (XL(NL) - XL(NL-1))
C              INITIALIZE CUBIC SPLINE INTERPOLATION
      CALL SPLN1(XL,YL,NL)
C              CALCULATE ORDINATES AND SLOPES
      IC = 0
      DO 330 I=ILE,ITE
      IC = IC + 1
      XP = XIN(I)
      CALL SPLN1X(XL,YL,NL)
C              SET ORDINATE AND SLOPE OF AIRFOIL TO INTERPOLATED
C              VALUE DIVIDED BY THICKNESS RATIO
      FL(IC) = YP* DELINV
      FXL(IC) = DYP*DELINV
  330 CONTINUE
      GO TO 500
  400 CONTINUE
C              BCFOIL = 4
C              THIS ADDRESS LEFT FOR USER TO INSERT OWN AIRFOIL
C              DESCRIPTION.
      WRITE(6,1000)
 1000 FORMAT(30H0ABNORMAL STOP IN SUBROUTINE BODY/
     1       24H BCTYPE=4 IS NOT USEABLE)
      STOP
  500 CONTINUE
C              EXECUTE  CALCULATIONS COMMON TO ALL AIRFOILS
C
C              COMPUTE AIRFOIL VOLUME
      CALL SIMP(VOLU,XFOIL,FU,IFOIL,IERR)
      CALL SIMP(VOLL,XFOIL,FL,IFOIL,IERR)
      VOL = VOLU - VOLL
C              COMPUTE CAMBER AND THICKNESS
      DO 525 I=1,IFOIL
      CAMBER(I) = .5*(FU(I) + FL(I))
      THICK(I) = .5*(FU(I) - FL(I))
  525 CONTINUE
      DO 530 I=1,IFOIL
      FXU(I) = FXU(I) / SQRT(1.0 + RIGF *(DELTA*FXU(I))**2 )
      FXL(I) = FXL(I) / SQRT(1.0 + RIGF *(DELTA*FXL(I))**2 )
  530 CONTINUE
      CALL PRBODY
      RETURN                                                            BODY   144
      END                                                               BODY   145




      SUBROUTINE CDCOLE                                                 CDCOLE   2
C               COMPUTES THE DRAG BY MOMENTUM INTEGRAL METHOD.          CDCOLE   3
C               INTEGRATES AROUND A CONTOUR ENCLOSING THE BODY AND      CDCOLE   4
C               ALONG ALL SHOCKS INSIDE THE CONTOUR                     CDCOLE   5
C               CALLED BY - PRINT.                                      CDCOLE   6
                                                                        CDCOLE   7
      COMMON          P(102,101),X(100)  , Y(100)                       BLANK    2
      COMMON / COM1/ IMIN    , IMAX    , IUP     , IDOWN   , ILE     ,  COM1     2
     1               ITE     , JMIN    , JMAX    , JUP     , JLOW    ,  COM1     3
     2               JTOP    , JBOT                                     COM1     4
      COMMON / COM2/ AK      , ALPHA   , DUB     , GAM1    , RTK        COM2     2
      COMMON / COM4/ FL(100) , FXL(100), FU(100) , FXU(100),            COM4     2
     1              CAMBER(100), THICK(100),VOL  , XFOIL(100), IFOIL    COM4     3
      COMMON / COM7/ CJUP    , CJUP1   , CJLOW   , CJLOW1               COM7     2
      COMMON /COM13/ CDFACT  , CLFACT  , CMFACT  , CPFACT  , CPSTAR     COM13    2
      LOGICAL        PHYS                                               COM27    2
      INTEGER        PRTFLO  , SIMDEF                                   COM27    3
      COMMON /COM27/ CL      , DELTA   , DELRT2  , EMACH   , EMROOT  ,  COM27    4
     1               PHYS    , PRTFLO  , SIMDEF  , SONVEL  , VFACT   ,  COM27    5
     2               YFACT                                              COM27    6
      COMMON/COM30/ XI(100), ARG(100), REST(204)                        CDCOLE  15
      GAM123 = GAM1*2./3.                                               CDCOLE  16
C               SET LOCATIONS OF CONTOUR BOUNDARIES                     CDCOLE  17
C                                                                       CDCOLE  18
C               UPSTREAM BOUNDARY                                       CDCOLE  19
C                         IF AK = 0.0 CDCOLE WILL NOT BE CALLED.        CDCOLE  20
C                            AMACH MAY NOT BE = 1.0 .                   CDCOLE  21
      IF(AK .GT. 0.) IU = (ILE + IMIN)*.5                               CDCOLE  22
      IF(AK .LT. 0.) IU = IUP                                           CDCOLE  23
C               TOP AND BOTTOM BOUNDARIES                               CDCOLE  24
C               SUBSONIC FREESTREAM                                     CDCOLE  25
C               SET JB,JT TO INCLUDE AS MUCH OF SHOCKS AS POSSIBLE      CDCOLE  26
      JT = JMAX - 1                                                     CDCOLE  27
      JB = JMIN + 1                                                     CDCOLE  28
      IF (AK .GT. 0.) GO TO 30                                          CDCOLE  29
C               SUPERSONIC FREESTREAM                                   CDCOLE  30
C               SET JB,JT TO INCLUDE ONLY SUBSONIC PART OF              CDCOLE  31
C               DETACHED BOW WAVE                                       CDCOLE  32
C                                                                       CDCOLE  33
C               FIND BOW SHOCK WAVE                                     CDCOLE  34
      ISTOP = ILE - 3                                                   CDCOLE  35
      CALL FINDSK(IUP,ISTOP,JUP,IBOW)                                   CDCOLE  36
      IF(IBOW .LT. 0) GO TO 325                                         CDCOLE  37
      IF (IBOW .LT. 4) GO TO 355                                        CDCOLE  38
C               SEARCH UP SHOCK TO FIND TIP OF SUBSONIC REGION          CDCOLE  39
      ISK = IBOW                                                        CDCOLE  40
      JSTART = JUP + 1                                                  CDCOLE  41
      JT = JUP - 1                                                      CDCOLE  42
      DO 10 J = JSTART,JMAX                                             CDCOLE  43
      JT = JT + 1                                                       CDCOLE  44
      ISKOLD = ISK                                                      CDCOLE  45
      CALL NEWISK(ISKOLD,J, ISK)                                        CDCOLE  46
      IF(ISK  .LT. 0) GO TO 15                                          CDCOLE  47
   10 CONTINUE                                                          CDCOLE  48
   15 CONTINUE                                                          CDCOLE  49
C               SEARCH DOWN SHOCK TO FIND TIP OF SUBSONIC REGION        CDCOLE  50
      ISK = IBOW                                                        CDCOLE  51
      JB = JLOW + 2                                                     CDCOLE  52
      DO 20 J = JMIN,JLOW                                               CDCOLE  53
      JJ = JLOW - J + JMIN                                              CDCOLE  54
      JB = JB - 1                                                       CDCOLE  55
      ISKOLD = ISK                                                      CDCOLE  56
      CALL NEWISK(ISKOLD,JJ,ISK)                                        CDCOLE  57
      IF(ISK .LT. 0) GO TO 25                                           CDCOLE  58
   20 CONTINUE                                                          CDCOLE  59
   25 CONTINUE                                                          CDCOLE  60
C               SAVE I LOCATION OF BOW SHOCK WAVE ON LOWER BOUNDARY     CDCOLE  61
      IBOW = ISKOLD                                                     CDCOLE  62
   30 CONTINUE                                                          CDCOLE  63
C               DOWNSTREAM BOUNDARY                                     CDCOLE  64
      ID = (ITE + IMAX) * .5                                            CDCOLE  65
      IF(PX(ITE+1,JUP) .LT. SONVEL)  GO TO 40                           CDCOLE  66
C               TRAILING EDGE IS SUPERSONIC.  PLACE DOWNSTREAM          CDCOLE  67
C               BOUNDARY AHEAD OF TRAILING EDGE TO AVOID TAIL SHOCK     CDCOLE  68
      I = ITE                                                           CDCOLE  69
   35 CONTINUE                                                          CDCOLE  70
      I = I-1                                                           CDCOLE  71
      IF(X(I) .GT. .75) GO TO 35                                        CDCOLE  72
      ID = I                                                            CDCOLE  73
   40 CONTINUE                                                          CDCOLE  74
C                                                                       CDCOLE  75
C                        ALL BOUNDARIES ARE FIXED                       CDCOLE  76
C                                                                       CDCOLE  77
C                        COMPUTE INTEGRALS ALONG BOUNDARIES             CDCOLE  78
C                                                                       CDCOLE  79
C                        INTEGRAL ON UPSTREAM BOUNDARY                  CDCOLE  80
      CDUP = 0.                                                         CDCOLE  81
      IF(AK .LT. 0.) GO TO 120                                          CDCOLE  82
      L = 0                                                             CDCOLE  83
      DO 110 J = JB,JT                                                  CDCOLE  84
      L = L+1                                                           CDCOLE  85
      XI(L) = Y(J)                                                      CDCOLE  86
      U = PX(IU,J)                                                      CDCOLE  87
      V = PY(IU,J)                                                      CDCOLE  88
      ARG(L) =((AK - GAM123*U)*U*U - V*V)*.5                            CDCOLE  89
  110 CONTINUE                                                          CDCOLE  90
      CALL TRAP(XI,ARG,L,SUM)                                           CDCOLE  91
      CDUP = 2.*CDFACT*SUM                                              CDCOLE  92
  120 CONTINUE                                                          CDCOLE  93
C                        INTEGRAL ON TOP BOUNDARY                       CDCOLE  94
      L = 0                                                             CDCOLE  95
      DO 130 I = IU,ID                                                  CDCOLE  96
      L = L + 1                                                         CDCOLE  97
      XI(L) = X(I)                                                      CDCOLE  98
      ARG(L) = -PX(I,JT)*PY(I,JT)                                       CDCOLE  99
  130 CONTINUE                                                          CDCOLE 100
      CALL TRAP(XI,ARG,L,SUM)                                           CDCOLE 101
      CDTOP = 2.*CDFACT*SUM                                             CDCOLE 102
C                        INTEGRAL ON BOTTOM BOUNDARY                    CDCOLE 103
      L = 0                                                             CDCOLE 104
       DO 140 I = IU,ID                                                 CDCOLE 105
      L = L +  1                                                        CDCOLE 106
      ARG(L) = PX(I,JB)*PY(I,JB)                                        CDCOLE 107
  140 CONTINUE                                                          CDCOLE 108
      CALL TRAP(XI,ARG,L,SUM)                                           CDCOLE 109
      CDBOT = 2.*CDFACT*SUM                                             CDCOLE 110
C                        INTEGRAL ON DOWNSTREAM BOUNDARY                CDCOLE 111
      L = 0                                                             CDCOLE 112
      DO 150 J = JB,JT                                                  CDCOLE 113
      L = L + 1                                                         CDCOLE 114
      XI(L) = Y(J)                                                      CDCOLE 115
      U = PX(ID,J)                                                      CDCOLE 116
C                        IF FLOW SUPERSONIC, USE BACKWARD DIFFERENCE FORMULA  CDCOLE 117
      IF(U .GT. SONVEL) U = PX(ID-1,J)                                  CDCOLE 118
      V = PY(ID,J)                                                      CDCOLE 119
      ARG(L) = ((GAM123*U - AK)*U*U + V*V)*.5                           CDCOLE 120
  150 CONTINUE                                                          CDCOLE 121
      CALL TRAP(XI,ARG,L,SUM)                                           CDCOLE 122
      CDDOWN = 2.*CDFACT*SUM                                            CDCOLE 123
C                        INTEGRAL ON BODY BOUNDARY                      CDCOLE 124
      CDBODY = 0.                                                       CDCOLE 125
      IF(ID .GT. ITE) GO TO 200                                         CDCOLE 126
      ILIM = ITE + 1                                                    CDCOLE 127
      L = 0                                                             CDCOLE 128
      DO 160 I=ID,ILIM                                                  CDCOLE 129
      IB = I-ILE + 1                                                    CDCOLE 130
      L = L + 1                                                         CDCOLE 131
      XI(L) = X(I)                                                      CDCOLE 132
      UU = CJUP*PX(I,JUP) - CJUP1*PX(I,JUP+1)                           CDCOLE 133
      UL = CJLOW*PX(I,JLOW) - CJLOW1*PX(I,JLOW-1)                       CDCOLE 134
      ARG(L) = -UU*FXU(IB) + UL*FXL(IB)                                 CDCOLE 135
  160 CONTINUE                                                          CDCOLE 136
      CALL TRAP(XI,ARG,L,SUM)                                           CDCOLE 137
      CDBODY = 2.*CDFACT*SUM                                            CDCOLE 138
  200 CONTINUE                                                          CDCOLE 139
C                                                                       CDCOLE 140
C                        INTEGRATION ALONG SHOCK WAVES                  CDCOLE 141
      CDWAVE = 0.                                                       CDCOLE 142
      LPRT1 = 0                                                         CDCOLE 143
      LPRT2 = 0                                                         CDCOLE 144
      NSHOCK = 0                                                        CDCOLE 145
      IF(AK .GT. 0.) GO TO 220                                          CDCOLE 146
C                        INTEGRATE ALONG DETACHED BOW WAVE              CDCOLE 147
      NSHOCK = NSHOCK + 1                                               CDCOLE 148
      LPRT1 = 1                                                         CDCOLE 149
      LPRT2 = 1                                                         CDCOLE 150
      L = 0                                                             CDCOLE 151
      ISK = IBOW                                                        CDCOLE 152
      DO 210 J = JB,JT                                                  CDCOLE 153
      L = L + 1                                                         CDCOLE 154
      ISKOLD = ISK                                                      CDCOLE 155
      CALL NEWISK(ISKOLD,J,ISK)                                         CDCOLE 156
      Z(L) = Y(J)
      ARG(L) = (PX(ISK+1,J) - PX(ISK-2,J))**3
  210 CONTINUE
      CALL TRAP(Z,ARG,L,SUM)
      CDSK = -GAM1/6.*CDFACT*SUM
      CDWAVE = CDWAVE + CDSK
      CALL PRTSK(Z,ARG,L,NSHOCK, CDSK,LPRT1)
  220 CONTINUE
C                    INTEGRATE ALONG SHOCKS ABOVE AIRFOIL
      ISTART = ILE
  225 CONTINUE
      CALL FINDSK(ISTART,ITE,JUP,ISK)
      IF(ISK .LT. 0) GO TO 250
C                    SHOCK WAVE FOUND
      ISTART = ISK + 1
      NSHOCK = NSHOCK + 1
      LPRT1 = 0
      L = 1
      Z(L) = 0.
      ARG(L) = (CJUP*(PX(ISK+1,JUP) - PX(ISK-2,JUP))
     *       - CJUP1*(PX(ISK+1,JUP+1) - PX(ISK-2,JUP+1)))**3
      DO 230 J = JUP , JT
      L = L+1
      Z(L) = Y(J)
      ARG(L) = (PX(ISK+1,J) - PX(ISK-2,J))**3
      ISKOLD = ISK
      JSK = J + 1
      CALL NEWISK(ISKOLD,JSK,ISK)
      IF(ISK .LT. 0) GO TO 240
      IF(ISK .GT. ID) GO TO 235
  230 CONTINUE
  235 CONTINUE
      LPRT1 = 1
  240 CONTINUE
      CALL TRAP(Z,ARG,L,SUM)
      CDSK = -GAM1/6.*CDFACT*SUM
      CDWAVE = CDWAVE + CDSK
      CALL PRTSK(Z,ARG,L,NSHOCK,CDSK,LPRT1)
      IF(LPRT1 .EQ. 1) LPRT2 = 1
C                    RETURN TO FIND NEXT SHOCK
      GO TO 225
C                    INTEGRATE ALONG SHOCKS BELOW AIRFOIL
  250 CONTINUE
      ISTART = ILE
  260 CONTINUE
      CALL FINDSK(ISTART,ITE,JLOW,ISK)
      IF(ISK .LT. 0) GO TO 300
C                    SHOCK WAVE FOUND
      ISTART = ISK + 1
      NSHOCK = NSHOCK + 1
      LPRT1 = 0
      L = 1
      Z(L) = 0.
      ARG(L) = (CJLOW*(PX(ISK+1,JLOW) - PX(ISK-2,JLOW))
     *       - CJLOW1*(PX(ISK+1,JLOW-1) - PX(ISK-2,JLOW-1)))**3
      DO 270 JJ = JB,JLOW
      J = JLOW + JB - JJ
      L = L+1
      Z(L) = Y(J)
      ARG(L) = (PX(ISK+1,J) - PX(ISK-2,J))**3
      ISKOLD = ISK
      JSK = J - 1
      CALL NEWISK(ISKOLD,JSK,ISK)
      IF(ISK .LT. 0) GO TO 280
      IF(ISK .GT. ID) GO TO 275
  270  CONTINUE
  275 CONTINUE
      LPRT1 = 1
  280 CONTINUE
      CALL TRAP(Z,ARG,L,SUM)
      CDSK = -GAM1/6.*CDFACT*(-SUM)
      CDWAVE = CDWAVE + CDSK
      CALL PRTSK(Z,ARG,L,NSHOCK,CDSK,LPRT1)
      IF(LPRT1 .EQ. 1) LPRT2 = 1
C                    RETURN TO FIND NEXT SHOCK
      GO TO 260
  300  CONTINUE
C
C                    INTEGRATION ALONG SHOCKS IS COMPLETE
C
C                    PRINTOUT CD INFORMATION
      XU = X(IU)
      XD = X(ID)
      YT = Y(JT)*YFACT
      YB = Y(JB)*YFACT
      CDC = CDUP + CDTOP + CDBOT + CDDOWN + CDBODY
      CD = CDC + CDWAVE
      WRITE(6,1001)
      WRITE(6,1002) XU,CDUP,  XD,CDDOWN,  YT,CDTOP,  YB,CDBOT
      IF(XD .LT. 1.)  WRITE(6,1003)  XD,CDBODY
      WRITE(6,1004) CDC
      WRITE(6,1005) NSHOCK,CDWAVE
      IF(NSHOCK .GT. 0 .AND. LPRT2 .EQ. 0)  WRITE(6,1007)
      IF(NSHOCK .GT. 0 .AND. LPRT2 .EQ. 1)  WRITE(6,1008)
      WRITE(6,1006) CD
      RETURN
  325 CONTINUE
C                    SHOCK IS TOO CLOSE TO BODY TO DO CONTOUR INTEGRAL.
C                    WRITE MESSAGE AND RETURN
      ULE = PX(ILE,JUP)
      IF(ULE .GT. SONVEL)  WRITE(6,1011)
      IF(ULE .LE. SONVEL)  WRITE(6,1012)
      CD = DRAG(CDFACT)
      WRITE(6,1013) CD
      RETURN
  330 WRITE (6,1014)
      CD=DRAG(CDFACT)
      WRITE(6,1013) CD
      RETURN
 1001 FORMAT(1H1,35X,23HDRAG COEFFICIENT OUTPUT/ 56X,23(1H*)//,
     1 60H CALCULATION OF DRAG COEFFICIENT BY MOMENTUM INTEGRAL METHOD)
 1002 FORMAT(//30HBOUNDARIES OF CONTOUR USED,15X,19HCONTRIBUTION TO CD/
     * 16H UPSTREAM     X =,F12.6,15X,8HCDUP    =,F12.6/
     * 16H DOWNSTREAM   X =,F12.6,15X,8HCDDOWN  =,F12.6/
     * 16H TOP          Y =,F12.6,15X,8HCDTOP   =,F12.6/
     * 16H BOTTOM       Y =,F12.6,15X,8HCDBOT   =,F12.6)
 1003 FORMAT(16H BODY AFT OF X =,F12.6,15X,8HCDBODY =,F12.6)
 1004 FORMAT(15X,36HTOTAL CONTRIBUTIONS AROUND CONTOUR =,F12.6)
 1005 FORMAT(10H0THERE ARE,I3,38H SHOCKS INSIDE CONTOUR. TOTAL CDWAVE =,
     *    F12.6)
 1006 FORMAT(51H0DRAG CALCULATED FROM MOMENTUM INTEGRAL    CD     =,
     *   F12.6)
 1007 FORMAT(48H0NOTE - ALL SHOCKS CONTAINED WITHIN CONTOUR/
     *  30H CDWAVE EQUALS TOTAL WAVE DRAG)
 1008 FORMAT(52H0NOTE - ONE OR MORE SHOCKS EXTEND OUTSIDE OF CONTOUR/
     *    39H CDWAVE DOES NOT EQUAL TOTAL WAVE DRAG)
 1011 FORMAT(31H1SHOCK WAVE IS ATTACHED TO BODY/
     *    33H MOMENTUM INTEGRAL CANNOT BE DONE/
     *    45H DRAG OBTAINED FROM SURFACE PRESSURE INTEGRAL/)
 1012 FORMAT(41H1DETACHED SHOCK WAVE IS TOO CLOSE TO BODY/
     *    33H MOMENTUM INTEGRAL CANNOT BE DONE/
     *    45H DRAG OBTAINED FROM SURFACE PRESSURE INTEGRAL/)
 1013 FORMAT(4H0CD=,F12.6)
 1014 FORMAT(1H1,17X,19H****** CAUTION *****/16X,16HSOLUTION MAY BE ,
     1 7HINVALID//46H DETACHED SHOCK WAVE IS TOO CLOSE TO UPSTREAM ,
     2 8HBOUNDARY/11X,32HMOMENTUM INTEGRAL CANNOT BE DONE/5X,
     3 44HDRAG OBTAINED FROM SURFACE PRESSURE INTEGRAL)
      END
```

```
      SUBROUTINE CKMESH

C                         CHECK X MESH AND ADJUST TO CONTAIN ODD NO. OF
C                         POINTS BEFORE TAIL AND ODD NO. AFTER TAIL.
C                         ITE IS INCLUDED IN BOTH COUNTS.
C                         CKMESH IS CALLED ONCE BY READIN.

      COMMON / COM1/ IMIN     , IMAX    , IUP     , IDOWN   , ILE     ,
     1               ITE      , JMIN    , JMAX    , JUP     , JLOW    ,
     2               JTOP     , JBOT
      LOGICAL        ABORT
      COMMON / COM3/ IREF     , ABORT   , ICUT    , KSTEP
      LOGICAL        AMESH
      COMMON / COM4/ XIN(100) , YIN(100), AMESH

      IF (ICUT .GT. 0) GO TO 4
      IREF = -1
      RETURN
    4 CONTINUE

C                         TEST TO BE SURE THAT ADJUSTING THE NO. OF
C                         POINTS WONT MAKE IMAX OR JMAX LARGER THAN 100.
      IF (IMAX .LE. 98 .AND. JMAX .LE. 98) GO TO 8
      WRITE (6,100)
      IREF = -1
      RETURN
    8 CONTINUE
      IF (MOD(ITE-IMIN+1, 2) .NE. 0) GO TO 20
C                             ADD EXTRA MESH POINT AHEAD OF AIRFOIL.
      LP = IMAX + IMIN + 1
      DO 10 I=IMIN,IMAX
      L = LP - I
      XIN(L) = XIN(L-1)
   10 CONTINUE
      IMAX = IMAX + 1
      XIN(IMIN) = 2. * XIN(IMIN+1) - XIN(IMIN+2)
      CALL ISLIT ( XIN )
   20 CONTINUE
C                             ADD EXTRA MESH POINT AFTER AIRFOIL.
      IF (MOD(IMAX-ITE+1, 2) .NE. 0) GO TO 30
      IMAX = IMAX + 1
      XIN(IMAX) = 2. * XIN(IMAX-1) - XIN(IMAX-2)
   30 CONTINUE
C                             CHECK Y MESH AND ADJUST TO CONTAIN EVEN NO. OF
C                             POINTS ABOVE AND BELOW SLIT.
      IF (MOD(JLOW-JMIN ,2) .NE. 0) GO TO 50
C                             ADD EXTRA MESH POINT BELOW SLIT.
      LP = JMAX + JMIN + 1
      DO 40 J=JMIN,JMAX
      L = LP - J
      YIN(L) = YIN(L-1)
   40 CONTINUE
      JMAX = JMAX + 1
      YIN(JMIN) = 2. * YIN(JMIN+1) - YIN(JMIN+2)
      CALL JSLIT ( YIN )
   50 CONTINUE
C                             ADD EXTRA MESH POINT ABOVE SLIT.
      IF (MOD(JMAX-JUP  ,2) .NE. 0) GO TO 60
      JMAX = JMAX + 1
      YIN(JMAX) = 2.0 * YIN(JMAX-1) - YIN(JMAX-2)
   60 CONTINUE
      RETURN

  100 FORMAT(49H0 THE MESH CANNOT BE ADJUSTED FOR CUTOUT, BECAUSE IMAX O
     1R JMAX IS TOO CLOSE TO THE LIMIT OF 100./
     25X,10HIREF WAS SET TO  -1 )

      END




      SUBROUTINE CPPLOT (X, Y, Z, W, NP)
C                             SUBROUTINE CPPLOT PRODUCES A PRINTER PLOT
C                             OF CRITICAL PRESSURE VS X .
C                             CALLED BY - FIXPLT.
      LOGICAL        AMESH
      COMMON / COM4/ XIN(100) , YIN(100), AMESH

      DIMENSION  X(101) , Y(101) , Z(101) , W(101) , M(120), ISYM(8)
      DIMENSION  A(3) , IC(3)
      DATA  IC/ 1, 1024, 1048576/
      DATA ISYM/1H ,1HU,1HL,1HB,1H-,1HU,1HL,1HB/
C                             NC IS THE NUMBER OF COLUMNS.
C                             NR IS THE NUMBER OF ROWS.
      DATA  NC /120/ , NR /50/
C     INITIALIZE RANGES
      IF ( AMESH ) GO TO 3
      NPL = 1
      NPR = NP - 1
      NLS = 2
      GO TO 4
    3 CONTINUE
      NPL = 2
      NPR = NP - 2
      NLS = 3
    4 CONTINUE
      HL = X(NPL)
      HR = X(NPL)
      VB=AMIN1(Y(1),Z(1),W(1))
      VT=AMAX1(Y(1),Z(1),W(1))
C     DETERMINE RANGES
      DO 5 I = NLS,NPR
      HL=AMIN1(HL,X(I))
      HR=AMAX1(HR,X(I))
      VB=AMIN1(VB,Y(I),Z(I),W(I))
      VT=AMAX1(VT,Y(I),Z(I),W(I))
 5    CONTINUE

C                             SKIP TO NEW PAGE AND WRITE PLOT HEADING.

      WRITE(6,900)
      VDEL=(VT-VB)/FLOAT(NR)
      HDEL=(HR-HL)/FLOAT(NC)
      HDELM = 1.0 / HDEL
      VL=VT


      DO 100 IROW = 1,NR
      VH=VL
      VL=FLOAT(NR-IROW)*VDEL+VB
      DO 15 I=1,NC
 15   M(I)=0

      DO 30 I =NPL,NPR
      J      = MAX0(1,MIN0(NC,1+INT((X(I)-HL) * HDELM)))
      A(1) = Y(I)
      A(2) = Z(I)
      A(3) = W(I)
      DO 20 K=1,3
      IF (A(K) .GT. VH) GO TO 20
      IF (A(K) .GT. VL .OR. (A(K) .LE. VB .AND. IROW .EQ. NR))
     1          M(J) = M(J) + IC(K)
   20 CONTINUE
   30 CONTINUE

      DO 90 I=1,NC
      J = 1
      IF (M(I) .LT. IC(3)) GO TO 70
      J = J + 4
      M(I) = MOD(M(I),IC(3))
   70 CONTINUE
      IF (M(I) .LT. IC(2)) GO TO 75
      J = J + 2
      M(I) = MOD(M(I),IC(2))
   75 CONTINUE
      IF (M(I) .LE. 0) GO TO 80
      J = J + 1
   80 CONTINUE
      M(I) = ISYM(J)
   90 CONTINUE

      WRITE(6,901)   (M(I),I=1,NC)
 100  CONTINUE

      RETURN

  900 FORMAT(1H1,34X,46HPRINTER PLOT OF CP ON BODY AND DIVIDING STREAM,
     1 4HLINE,15X,13HU = CP(UPPER)/35X,50(1H-),15X,13HL = CP(LOWER)/
     2 100X,25HB = CP(UPPER) = CP(LOWER)/90X,17H--- = CRITICAL CP)
  901 FORMAT(1X,120A1)

      END




      SUBROUTINE CUTOUT
C                             SUBROUTINE TO REDUCE THE NUMBER OF MESH POINTS
C                             FOR THE FIRST CUT AT SOLUTION. THE X-MESH AND
C                             Y-MESH WILL BE HALVED, AND IF POSSIBLE BE
C                             HALVED AGAIN.
C                                  IREF =-1, CUTOUT SHOULDNT HAVE BEEN CALLED
C                                  IREF = 0, IF NOT HALVED AT ALL.
C                                  IREF = 1, IF HALVED ONCE.
C                                  IREF = 2, IF HALVED TWICE.
C                             CALLED BY - TSFOIL.

      COMMON         P(102,101),X(100)  , Y(100)
      COMMON / COM1/ IMIN     , IMAX    , IUP     , IDOWN   , ILE     ,
     1               ITE      , JMIN    , JMAX    , JUP     , JLOW    ,
     2               JTOP     , JBOT
      LOGICAL        ABORT
      COMMON / COM3/ IREF     , ABORT   , ICUT    , KSTEP
      LOGICAL        AMESH
      COMMON / COM4/ XIN(100) , YIN(100), AMESH
      COMMON /COM20/ XMID(100), YMID(100)
      IF(IREF .NE. -1) GO TO 5
C                   MESH CANNOT BE REFINED. LOAD XIN,YIN INTO X,Y
      DO 1 I = IMIN,IMAX
      X(I) = XIN(I)
 1    CONTINUE
      DO 2 J = JMIN,JMAX
      Y(J) = YIN(J)
 2    CONTINUE
      IREF = 0
      RETURN
 5    CONTINUE
      K = IMIN - 1
      DO 10 I=IMIN,IMAX,2
      K = K + 1
      XMID(K) = XIN(I)
   10 CONTINUE
      IMAX = (IMAX-IMIN) * .5 + IMIN
      CALL ISLIT ( XMID )
      K = JMIN - 1
      JE = JLOW - 1
      DO 15 J=JMIN,JE,2
      K = K + 1
      YMID(K) = YIN(J)
   15 CONTINUE
      JST = JUP + 1
      DO 20 J=JST,JMAX,2
      K = K + 1
      YMID(K) = YIN(J)
   20 CONTINUE
      JMAX = (JMAX-JMIN) * .5 + JMIN
      CALL JSLIT ( YMID )
      IREF = 1
C                            FIRST HALVING COMPLETE. CHECK IF NO. OF POINTS
C                            IS ODD.
      IF (ICUT .EQ. 1) GO TO 30
      IF (MOD(ITE-IMIN+1, 2) .EQ. 0) GO TO 30
      IF (MOD(IMAX-ITE+1,2) .EQ. 0) GO TO 30
      IF (MOD(JLOW-JMIN ,2) .EQ. 0) GO TO 30
      IF (MOD(JMAX-JUP  ,2) .NE. 0) GO TO 60
   30 CONTINUE
C                            ONLY ONE MESH REFINEMENT POSSIBLE.
      DO 40 I=IMIN,IMAX
      X(I) = XMID(I)
   40 CONTINUE
      DO 50 J=JMIN,JMAX
      Y(J) = YMID(J)
   50 CONTINUE
      RETURN
C                            ALL POINTS ARE ODD SO CUT AGAIN.
   60 CONTINUE
      K = IMIN - 1
      DO 70 I=IMIN,IMAX,2
      K = K + 1
      X(K) = XMID(I)
   70 CONTINUE
      IMAX = (IMAX-IMIN) * .5 + IMIN
      CALL ISLIT ( X )
      K = JMIN - 1
      JE = JLOW - 1
      DO 75 J=JMIN,JE,2
      K = K + 1
      Y(K) = YMID(J)
   75 CONTINUE
      JST = JUP + 1
      DO 80 J=JST,JMAX,2
      K = K + 1
      Y(K) = YMID(J)
   80 CONTINUE
      JMAX = (JMAX-JMIN) * .5 + JMIN
      CALL JSLIT ( Y )
      IREF = 2
      RETURN

      END
```

```
      SUBROUTINE DIFCOE
C
C  COMPUTE DIFFERENCE COEFFICIENTS IN FIELD
C                  CALLED BY - TSFOIL.
C
      COMMON          P(102,101),X(100)  , Y(100)
      COMMON / COM1/ IMIN     , IMAX    , IUP     , IDOWN   , ILE     ,
     1               ITE      , JMIN    , JMAX    , JUP     , JLOW    ,
     2               JTOP     , JBOT
      COMMON / COM2/ AK       , ALPHA   , DUB     , GAM1    , RTK
      COMMON / COM5/ XDIFF(100),YDIFF(100)
      COMMON / COM7/ CJUP     , CJUP1   , CJLOW   , CJLOW1
      COMMON /COM17/ CYYBLC   , CYYBLD  , CYYBLU  , CYYBUC  , CYYBUD  ,
     1               CYYBUU   ,FXLBC(100),FXUBC(100)
      COMMON /COM22/ CXC(100) , CXL(100), CXR(100), CXXC(100),CXXL(100),
     1               CXXR(100), C1(100)
      COMMON /COM23/ CYYC(100), CYYD(100),CYYU(100), IVAL
C
C                           COEFFICIENTS FOR (PIX AND (PIXX AT IMIN
      CXXL(IMIN) = 0.0
      CXXR(IMIN) = 0.0
      CXXC(IMIN) = 0.0
      CXL (IMIN) = 0.0
      CXR (IMIN) = 0.0
      CXC (IMIN) = 0.0
C  COEFFICIENTS FOR (PIX AND (PIXX FROM I=IMIN+1 TO I=IMAX-1
      C2 = GAM1*.5
      ISTART = IMIN+1
      IEND=IMAX-1
      DO 1 I=ISTART,IEND
      DXL=X(I)-X(I-1)
      DXR=X(I+1)-X(I)
      DXC = .5 * (X(I+1) - X(I-1))
C  FOR VC
      C1(I)=    AK/DXC
C  FOR (PIX
      CXL(I) = -C2 / (DXL*DXC)
      CXR(I) =  C2 / (DXR*DXC)
      CXC(I) = -CXL(I) - CXR(I)
C  FOR (PIXX
      CXXL(I) = 1.0 / DXL
      CXXR(I) = 1.0 / DXR
      CXXC(I) = CXXL(I) + CXXR(I)
 1    CONTINUE
C                  COEFFICIENTS FOR (PIX AND (PIXX AT IMAX
      DX             = X(IMAX) - X(IMAX-1)
      Q = 1.0 / (DX*DX)
      C1(IMAX) = AK / DX
      CXL(IMAX)  = -C2 * Q
      CXR(IMAX)  =  C2 * Q
      CXC(IMAX) = 0.0
      CXXL(IMAX) =  1.0 / DX
      CXXR(IMAX) =  1.0 / DX
      CXXC(IMAX) =  CXXL(IMAX) + CXXR(IMAX)
C                                COEFFICIENTS FOR (PIYY AT JMIN.
C
      DYU = Y(JMIN+1) - Y(JMIN)
      CYYD(JMIN) = 2.0 / DYU
      CYYU(JMIN) = 2./(DYU*DYU)
      CYYC(JMIN) = CYYU(JMIN)
C  COEFFICIENTS FOR (PIYY FROM J=JMIN+1 TO J=JMAX-1
      JSTART = JMIN + 1
      JEND=JMAX-1
      DO 2 J=JSTART,JEND
      DYD=Y(J)-Y(J-1)
      DYU=Y(J+1)-Y(J)
      DYC=Y(J+1)-Y(J-1)
      CYYD(J)=2./(DYD*DYC)
      CYYU(J)=2./(DYU*DYC)
      CYYC(J) = CYYD(J) + CYYU(J)
 2    CONTINUE
C                               COEFFICIENTS FOR (PIYY AT JMAX.
      DYD = Y(JMAX) - Y(JMAX-1)
      CYYD(JMAX) = 2./(DYD*DYD)
      CYYU(JMAX) = 2./DYD
      CYYC(JMAX) = CYYD(JMAX)

C  COEFFICIENTS FOR VELOCITY FORMULAS

      ISTART = IMIN + 1
      DO 3 I = ISTART,IMAX
      XDIFF(I) = 1./(X(I) - X(I-1))
 3    CONTINUE
      JSTART = JMIN +1
      DO 5 J = JSTART , JMAX
      YDIFF(J) = 1./(Y(J) - Y(J-1))
```

```
    5 CONTINUE

C COEFFICIENTS FOR EXTRAPOLATION FORMULAS FOR AIRFOIL SURFACE PROPERTIES

      CJLOW  = -Y(JLOW-1) / (Y(JLOW) - Y(JLOW-1))
      CJLOW1 = -Y(JLOW)   / (Y(JLOW) - Y(JLOW-1))
      CJUP   = Y(JUP+1) / (Y(JUP+1) - Y(JUP))
      CJUP1  = Y(JUP)   / (Y(JUP+1) - Y(JUP))
C                  COMPUTE SPECIAL DIFFERENCE COEFFICIENTS FOR PYY
C                  TO USE FOR AIRFOIL BOUNDARY CONDITION
C                  DIFFERENCE COEFFICIENTS FOR UPPER SURFACE
      CYYBUD = -2./(Y(JUP+1) + Y(JUP))
      CYYBUC = -CYYBUD / (Y(JUP+1) - Y(JUP))
      CYYBUU =  CYYBUC
C                  DIFFERENCE COEFFICIENTS FOR LOWER SURFACE
      CYYBLU=-2.0/(Y(JLOW) + Y(JLOW-1))
      CYYBLC = CYYBLU / (Y(JLOW) - Y(JLOW-1))
      CYYBLD = CYYBLC
      RETURN
      END
```

```
      FUNCTION DRAG(CDFACT)
C                  COMPUTES PRESSURE DRAG BY INTEGRATING U*V AROUND
C                  AIRFOIL USING TRAPEZOIDAL RULE.
C                  CALLED BY - CDCOLE.

      COMMON         P(102,101),X(100)  , Y(100)
      COMMON / COM1/ IMIN     , IMAX    , IUP     , IDOWN   , ILE     ,
     1               ITE      , JMIN    , JMAX    , JUP     , JLOW    ,
     2               JTOP     , JBOT
      COMMON / COM6/ FL(100)  , FXL(100), FU(100) , FXU(100),
     1          CAMBER(100), THICK(100),VOL    , XFOIL(100), IFOIL
      COMMON / COM7/ CJUP     , CJUP1   , CJLOW   , CJLOW1
      COMMON /COM30/ XI(100)  , ARG(100)  , REST(204)

      K = 1
      ARG(1) = 0.
      XI(1) = X(ILE-1)
      DO 10 I = ILE,ITE
      K = K+1
      PXUP = CJUP*PX(I,JUP) - CJUP1*PX(I,JUP+1)
      PXLOW = CJLOW*PX(I,JLOW) - CJLOW1*PX(I,JLOW-1)
      ARG(K) = FXU(K-1)*PXUP - FXL(K-1)*PXLOW
      XI(K) = X(I)
   10 CONTINUE
      K = K + 1
      ARG(K) = 0.
      XI(K) = X(ITE+1)
      CALL TRAP(XI,ARG,K,SUM)
      DRAG = - SUM*CDFACT*2.
      RETURN
      END
```

```
      SUBROUTINE DROOTS
C                  COMPUTE CONSTANTS ALPHA0,ALPHA1,ALPHA2,OMEGA0,OMEGA1
C                  OMEGA2 USED IN FORMULA FOR DOUBLET IN SLOTTED WIND
C                  TUNNEL WITH SUBSONIC FREESTREAM
C                  CALLED BY - FARFLD.

      COMMON /COM12/ F        , H         , HALFPI  , PI        , RTKPOR  ,
     1               TWOPI
      REAL JET
      COMMON /COM16/ ALPHA0   , ALPHA1  , ALPHA2  , XSING     ,
     1               OMEGA0   , OMEGA1  , OMEGA2  , JET

      ERROR = .00001
C                  COMPUTE ALPHA0
      ALPHA0 = 0.
      DO 10 I = 1,100
      TEMP = ALPHA0
      Q = F*TEMP - RTKPOR
      ALPHA0 = HALFPI - ATAN(Q)
      DALPHA = ABS(ALPHA0 - TEMP)
      IF(DALPHA .LT. ERROR) GO TO 15
   10 CONTINUE
      N = 0
      GO TO 9999
   15 CONTINUE
C                  COMPUTE ALPHA1
      ALPHA1 = 0.
      DO 20 I = 1,100
      TEMP = ALPHA1
      Q = F*(TEMP - PI) - RTKPOR
      ALPHA1 = HALFPI - ATAN(Q)
      DALPHA = ABS(ALPHA1 - TEMP)
      IF(DALPHA .LT. ERROR) GO TO 25
   20 CONTINUE
      N = 1
      GO TO 9999
   25 CONTINUE
C                  COMPUTE ALPHA2
      ALPHA2 = 0.
      DO 30 I=1,100
      TEMP = ALPHA2
      Q = F*(TEMP - TWOPI) - RTKPOR
      ALPHA2 = HALFPI - ATAN(Q)
      DALPHA = ABS(ALPHA2 - TEMP)
      IF(DALPHA .LT. ERROR) GO TO  35
   30 CONTINUE
      N = 2
      GO TO 9999
   35 CONTINUE
C                  COMPUTE OMEGA0,OMEGA1,OMEGA2
      TEMP = 1.0 / TAN(ALPHA0)
      OMEGA0 = 1./(1. + F/(1.+ TEMP*TEMP))
      TEMP = 1.0 / TAN(ALPHA1)
      OMEGA1 = 1./(1. + F/(1. + TEMP*TEMP))
      TEMP = 1.0 / TAN(ALPHA2)
      OMEGA2 = 1./(1. + F/(1. + TEMP*TEMP))
      RETURN
C                  ABNORMAL STOP IF ITERATION FOR ALPHAS NOT CONVERGED
 9999 CONTINUE
      WRITE(6,1000) N
 1000 FORMAT(35H1ABNORMAL STOP IN SUBROUTINE DROOTS/
     1  30H0NONCONVERGENCE OF ITERATION FOR ALPHA, I1)
      STOP
      END
```

```
      SUBROUTINE ECHINP
C                  PRINTS INPUT CARDS USED FOR RUN.
C                  CALLED BY - TSFOIL.
      DIMENSION   CRD(8)
      WRITE (6,110)
   10 CONTINUE
      READ (5,100) CRD
C                YES, NO
      IF (EOF(5)) 30,20
   20 CONTINUE
      WRITE (6,101) CRD
      GO TO 10
   30 CONTINUE
      REWIND 5
      RETURN

  100 FORMAT(8A10)
  101 FORMAT(1X,8A10)
  110 FORMAT(1H1,48X,35HLISTING OF INPUT CARDS FOR THIS RUN/48X,35(1H=)
     1  ///)

      END
      FUNCTION EMACH1(U)                                                EMACH1   2
C               FUNCTION EMACH1 COMPUTES LOCAL SIMILARITY PARAMETER     EMACH1   3
C               OR LOCAL MACH NUMBER                                    EMACH1   4
C               CALLED BY - MACHMP, PRINT1, PRTFLD.                     EMACH1   5
                                                                        EMACH1   6
      COMMON / COM2/ AK       , ALPHA  , DUB    , GAM1   , RTK          COM2     2
      LOGICAL        PHYS                                               COM27    2
      INTEGER        PRTFLD  , SIMDEF                                   COM27    3
      COMMON /COM27/ CL       , DELTA  , DELRT2 , EMACH  , EMROOT ,     COM27    4
     1               PHYS     , PRTFLD , SIMDEF , SONVEL , VFACT  ,     COM27    5
     2               YFACT                                              COM27    6
                                                                        EMACH1   9
C               COMPUTE SIMILARITY PARAMETER BASED ON LOCAL VELOCITY    EMACH1  10
      AK1 = AK - GAM1*U                                                 EMACH1  11
      IF(PHYS) GO TO 5                                                  EMACH1  12
C               RETURN VALUE OF LOCAL SIMILARITY PARAMETER              EMACH1  13
      EMACH1 = AK1                                                      EMACH1  14
      RETURN                                                            EMACH1  15
    5 CONTINUE                                                          EMACH1  16
C               COMPUTE VALUE OF LOCAL MACH NUMBER AND RETURN           EMACH1  17
C               COLE SCALING                                            EMACH1  18
      ARG = DELRT2*AK1                                                  EMACH1  19
C               SPREITER SCALING                                        EMACH1  20
      IF(SIMDEF .EQ. 2) ARG = ARG*EMROOT*EMROOT                         EMACH1  21
C               KRUPP SCALING                                           EMACH1  22
      IF(SIMDEF .EQ. 3)  ARG = ARG*EMACH                                EMACH1  23
      ARG = 1. - ARG                                                    EMACH1  24
      EMACH1 = 0.                                                       EMACH1  25
      IF(ARG .GT. 0.)EMACH1 = SQRT(ARG)                                 EMACH1  26
      RETURN                                                            EMACH1  27
      END                                                               EMACH1  28
```

```
      SUBROUTINE EXTRAP(XP,YP,PNEW)                                     EXTRAP   2
C               COMPUTE P AT X, YP USING FAR FIELD SOLUTION             EXTRAP   3
C               FOR SUBSONIC FLOW                                       EXTRAP   4
C               CALLED BY - GUESSP.                                     EXTRAP   5
                                                                        EXTRAP   6
      COMMON / COM2/ AK       , ALPHA  , DUB    , GAM1   , RTK          COM2     2
      COMMON /COM12/ F        , H      , HALFPI , PI     , RTKPOR ,     COM12    2
     1               TWOPI                                              COM12    3
      COMMON /COM15/ B        , BETA0  , BETA1  , BETA2  , PSI0   ,     COM15    2
     1               PSI1     , PSI2                                    COM15    3
      REAL JET                                                          COM16    2
      COMMON /COM16/ ALPHA0   , ALPHA1 , ALPHA2 , XSING  ,              COM16    3
     1               OMEGA0   , OMEGA1 , OMEGA2 , JET                   COM16    4
      INTEGER        BCTYPE                                             COM26    2
      COMMON /COM26/ BCTYPE   , CIRCFF , FHINV  , POR    , CIRCTE       COM26    3
                                                                        EXTRAP  12
      IF(BCTYPE .NE. 1) GO TO 100                                       EXTRAP  13
C               FREE AIR BOUNDARY CONDITION                             EXTRAP  14
      IF(ABS(YP) .LT. 1.E-6)  YP = -1.E-6                               EXTRAP  15
      XI = XP - XSING                                                   EXTRAP  16
      ETA = YP*RTK                                                      EXTRAP  17
      PNEW = - CIRCFF/TWOPI*(ATAN2(ETA,XI) + PI - SIGN(PI,ETA))         EXTRAP  18
     1       + DUB /TWOPI/RTK*(XI/(XI*XI + ETA*ETA))                    EXTRAP  19
      RETURN                                                            EXTRAP  20
  100 CONTINUE                                                          EXTRAP  21
C               TUNNEL WALL BOUNDARY CONDITION                          EXTRAP  22
      ETA = YP/H                                                        EXTRAP  23
      XI = (XP - XSING)/(H*RTK)                                         EXTRAP  24
      IF(XI .LT. 0.) GO TO 200                                          EXTRAP  25
C               XP IS DOWNSTREAM OF AIRFOIL                             EXTRAP  26
      TERM = ETA                                                        EXTRAP  27
      IF(BCTYPE .NE. 3)  TERM = SIN(ETA*BETA0)/BETA0                    EXTRAP  28
      PNEW = - .5*CIRCFF*(1. - SIGN(1.,ETA)                             EXTRAP  29
     1               + (1.-JET)*PSI0*TERM*EXP(-BETA0*XI))               EXTRAP  30
     2       + DUB *.5/(AK*H)*(B + OMEGA0*COS(ETA*ALPHA0)               EXTRAP  31
     3               *EXP(-ALPHA0*XI))                                  EXTRAP  32
      RETURN                                                            EXTRAP  33
  200 CONTINUE                                                          EXTRAP  34
C               XP IS UPSTREAM OF AIRFOIL                               EXTRAP  35
      TERM = 0.                                                         EXTRAP  36
      IF(JET .NE. 0.) TERM = JET*ETA/(1.+F)                             EXTRAP  37
      ARG1 = PI - ALPHA1                                                EXTRAP  38
      ARG2 = PI - BETA2                                                 EXTRAP  39
      PNEW = -.5*CIRCFF*(1.-TERM-PSI2*SIN(ETA*ARG2)/ARG2*EXP(ARG2*XI))  EXTRAP  40
     1       -.5*DUB /(AK*H)*((1.-B)*OMEGA1*COS(ETA*ARG1)*EXP(XI*ARG1)) EXTRAP  41
      RETURN                                                            EXTRAP  42
      END                                                               EXTRAP  43
```

```
      SUBROUTINE FARFLD                                                 FARFLD   2
C               SUBROUTINE COMPUTES BOUNDARY DATA FOR OUTER             FARFLD   3
C               BOUNDARIES.                                             FARFLD   4
C               CALLED BY - TSFOIL.                                     FARFLD   5
                                                                        FARFLD   6
      COMMON / COM1/ IMIN     , IMAX   , IUP    , IDOWN  , ILE    ,     COM1     2
     1               ITE      , JMIN   , JMAX   , JUP    , JLOW   ,     COM1     3
     2               JTOP     , JBOT                                    COM1     4
      COMMON / COM2/ AK       , ALPHA  , DUB    , GAM1   , RTK          COM2     2
      LOGICAL        AMESH                                              COM4     2
      COMMON / COM4/ XIN(100) , YIN(100), AMESH                         COM4     3
      COMMON /COM12/ F        , H      , HALFPI , PI     , RTKPOR ,     COM12    2
     1               TWOPI                                              COM12    3
      COMMON /COM15/ B        , BETA0  , BETA1  , BETA2  , PSI0   ,     COM15    2
     1               PSI1     , PSI2                                    COM15    3
      REAL JET                                                          COM16    2
      COMMON /COM16/ ALPHA0   , ALPHA1 , ALPHA2 , XSING  ,              COM16    3
     1               OMEGA0   , OMEGA1 , OMEGA2 , JET                   COM16    4
      COMMON /COM24/ DTOP(100), DBOT(100),DUP(100), DDOWN(100),         COM24    2
     1               VTOP(100), VBOT(100),VUP(100), VDOWN(100)          COM24    3
      INTEGER        BCTYPE                                             COM26    2
      COMMON /COM26/ BCTYPE   , CIRCFF , FHINV  , POR    , CIRCTE       COM26    3
C                                                                       FARFLD  15
C               TEST FOR SUPERSONIC OR SUBSONIC FREESTREAM              FARFLD  16
      IF(AK .GT. 0.) GO TO 99                                           FARFLD  17
      IF (F .NE. 0.0 .AND. H .NE. 0.0) GO TO 10                         FARFLD  18
      FHINV = 1.0                                                       FARFLD  19
      RETURN                                                            FARFLD  20
   10 CONTINUE                                                          FARFLD  21
      FHINV = 1.0 / (F * H)                                             FARFLD  22
C               SUPERSONIC FREESTREAM                                   FARFLD  23
C               UPSTREAM BOUNDARY CONDITIONS CORRESPOND TO UNIFORM      FARFLD  24
C               UNDISTURBED FLOW.                                       FARFLD  25
C               DOWNSTREAM BOUNDARY REQUIRED TO BE SUPERSONIC           FARFLD  26
C               TOP AND BOTTOM BOUNDARIES USE SIMPLE WAVE SOLUTION.     FARFLD  27
      RETURN                                                            FARFLD  28
C                                                                       FARFLD  29
   99 CONTINUE                                                          FARFLD  30
C                                                                       FARFLD  31
C               SUBSONIC FREESTREAM                                     FARFLD  32
C                                                                       FARFLD  33
C               FUNCTIONAL FORM OF THE POTENTIAL ON OUTER BOUNDARIES    FARFLD  34
C               IS PRESCRIBED. EQUATIONS REPRESENT ASYMPTOTIC FORM      FARFLD  35
C               FOR DOUBLET AND VORTEX IN FREE AIR AND WIND TUNNEL      FARFLD  36
C               ENVIRONMENT. DOUBLET AND VORTEX ARE LOCATED AT          FARFLD  37
C               X=XSING, Y=0.                                           FARFLD  38
C               ACTUAL BOUNDARY VALUES ARE SET IN SUBROUTINES RECIRC    FARFLD  39
C               AND REDUB WHERE THE FUNCTIONAL FORMS ARE MULTIPLIED     FARFLD  40
C               BY THE VORTEX AND DOUBLET STRENGTHS.  THE BOUNDARY      FARFLD  41
C               CONDITIONS ARE CALCULATED HEREIN FOR THE INPUT X AND    FARFLD  42
C               Y MESH AND VALUES ARE DELETED FOR THE COARSE MESH IN    FARFLD  43
C               SUBROUTINE SETBC.                                       FARFLD  44
                                                                        FARFLD  45
C               SET LOCATION OF SINGULAR VORTEX AND DOUBLET             FARFLD  46
      XSING = .5                                                        FARFLD  47
C               SET DEFAULT VALUES FOR TUNNEL WALL PARAMETERS           FARFLD  48
      B = 0.                                                            FARFLD  49
      OMEGA0 = 1.                                                       FARFLD  50
      OMEGA1 = 1.                                                       FARFLD  51
      OMEGA2 = 1.                                                       FARFLD  52
      JET = 0.                                                          FARFLD  53
      PSI0 = 1.                                                         FARFLD  54
      PSI1 = 1.                                                         FARFLD  55
      PSI2 = 1.                                                         FARFLD  56
C               BRANCH TO APPROPRIATE FORMULAS DEPENDING ON BCTYPE      FARFLD  57
      GO  TO (100,200,300,400,500,600),BCTYPE                           FARFLD  58
  100 CONTINUE                                                          FARFLD  59
C               BCTYPE = 1                                              FARFLD  60
C               FREE AIR BOUNDARY CONDITION                             FARFLD  61
C               SET BOUNDARY ORDINATES                                  FARFLD  62
      YT = YIN(JMAX)*RTK                                                FARFLD  63
      YB = YIN(JMIN)*RTK                                                FARFLD  64
      XU = XIN(IMIN) - XSING                                            FARFLD  65
      XD = XIN(IMAX) - XSING                                            FARFLD  66
      YT2 = YT*YT                                                       FARFLD  67
      YB2 = YB*YB                                                       FARFLD  68
      XU2 = XU*XU                                                       FARFLD  69
      XD2 = XD*XD                                                       FARFLD  70
      COEF1 = 1./TWOPI                                                  FARFLD  71
      COEF2 = 1.0/(TWOPI*RTK)                                           FARFLD  72
C               COMPUTE DOUBLET AND VORTEX TERMS ON TOP AND BOTTOM      FARFLD  73
C               BOUNDARIES.                                             FARFLD  74
      DO 110 I=IMIN,IMAX                                                FARFLD  75
      XP = XIN(I) - XSING                                               FARFLD  76
      XP2 = XP*XP                                                       FARFLD  77
      DTOP(I) = XP/(XP2 + YT2)*COEF2                                    FARFLD  78
      DBOT(I) = XP/(XP2 + YB2)*COEF2
      VTOP(I) = -ATAN2(YT,XP)*COEF1
      VBOT(I)  = -(ATAN2(YB,XP) + TWOPI)*COEF1
  110 CONTINUE
C                    COMPUTE DOUBLET AND VORTEX TERMS ON UPSTREAM
C                    AND DOWNSTREAM BOUNDARIES
      DO 120 J=JMIN,JMAX
      YJ = Y(J)*RTK
      YJ2 = YJ*YJ
      DUP(J) = XU/(XU2 + YJ2)*COEF2
      DDOWN(J) = XD/(XD2 + YJ2)*COEF2
      Q = PI - SIGN(PI,YJ)
      VUP(J) = -(ATAN2(YJ,XU) + Q)*COEF1
      VDOWN(J) = -(ATAN2(YJ,XD) + Q)*COEF1
  120 CONTINUE
      IF (AK .GT. 0.0) CALL ANGLE
      RETURN
C                    COMPUTE WALL CONSTANTS FOR VARIOUS WIND TUNNEL
C                    CONDITIONS.
  200 CONTINUE
C                    BCTYPE = 2
C                    SOLID WALL TUNNEL
      POR = 0.
C                    SET CONSTANTS FOR DOUBLET SOLUTION
      B = .5
      ALPHA0 = PI
      ALPHA1 = PI
      ALPHA2 = PI
C                    SET CONSTANTS FOR VORTEX SOLUTION
      BETA0 = HALFPI
      BETA1 = HALFPI
      BETA2 = HALFPI
      GO TO 700
  300 CONTINUE
C                    BCTYPE = 3
C                    FREE JET
      F = 0.
      RTKPOR = 0.
C                    SET CONSTANTS FOR DOUBLET SOLUTION
      ALPHA0 = HALFPI
      ALPHA1 = HALFPI
      ALPHA2 = HALFPI
C                    SET CONSTANTS FOR VORTEX SOLUTION
      JET = .5
      BETA0 = 0.
      BETA1 = 0.
      BETA2 = 0.
      GO TO 700
  400 CONTINUE
C                    BCTYPE = 4
C                    IDEAL SLOTTED WALL
      RTKPOR = 0.
      FHINV = 1.0 / ( F * H )
C                    SET CONSTANTS FOR DOUBLET SOLUTION
      CALL DROOTS
C                    SET CONSTANTS FOR VORTEX SOLUTION
      JET = .5
      CALL VROOTS
      GO TO 700
  500 CONTINUE
C                    BCTYPE = 5
C                    IDEAL PERFORATED/POROUS WALL
      F = 0.
      RTKPOR = RTK/POR
C                    SET CONSTANTS FOR DOUBLET SOLUTION
      ALPHA0 = HALFPI - ATAN(-RTKPOR)
      ALPHA1 = ALPHA0
      ALPHA2 = ALPHA0
C                    SET CONSTANTS FOR VORTEX SOLUTION
      BETA0 = ATAN(RTKPOR)
      BETA1 = BETA0
      BETA2 = BETA0
      GO TO 700
  600 CONTINUE
C                    BCTYPE = 6
C                    GENERAL HOMOGENEOUS WALL BOUNDARY CONDITION
C                    BOUNDARY CONDITION IS NOT OPERABLE YET IN FINITE
C                    DIFFERENCE SUBROUTINES.
C                    FAR FIELD SOLUTION HAS BEEN DERIVED AND IS INCLUDED
C                    HERE  FOR FUTURE USE
      RTKPOR = RTK/POR
      CALL DROOTS
      CALL VROOTS
      WRITE(6,1000)
 1000 FORMAT(35H1ABNORMAL STOP IN SUBROUTINE FARFLD/
     1       24H BCTYPE=6 IS NOT USEABLE)
      STOP
  700 CONTINUE
C                    COMPUTE FUNCTIONAL FORMS FOR UPSTREAM AND DOWNSTREAM
C                    BOUNDARY CONDITIONS  FOR DOUBLET AND VORTEX
      XU = (XIN(IMIN) - XSING)/(RTK*H)
      XD = (XIN(IMAX) - XSING)/(RTK*H)
C                    DOUBLET TERMS
      COEF1 = .5/AK/H
      ARG0 = ALPHA0
      ARG1 = PI - ALPHA1
      ARG2 = TWOPI - ALPHA2
      EXARG0 = EXP(-ARG0*XD)
      EXARG1 = EXP(ARG1*XU)
      EXARG2 = EXP(ARG2*XU)
      DO 720 J = JMIN,JMAX
      YJ = Y(J) / H
      DDOWN(J) = COEF1*(B + OMEGA0*COS(YJ*ARG0)*EXARG0)
      DUP(J) = -COEF1*((1.-B)*OMEGA1*COS(YJ*ARG1)*EXARG1 +
     1  OMEGA2*COS(YJ*ARG2)*EXARG2)
  720 CONTINUE
C                    VORTEX TERMS
      ARG0 = BETA0
      ARG1 = PI + BETA1
      ARG2 = PI - BETA2
      EXARG0 = EXP(-ARG0*XD)
      EXARG1 = EXP(-ARG1*XD)
      EXARG2 = EXP(ARG2*XU)
      DO 740 J = JMIN,JMAX
      YJ = Y(J) / H
      TERM = YJ
      IF (JET .EQ. 0.0) TERM = SIN(YJ*ARG0) / ARG0
      VDOWN(J) = -.5*(1. - SIGN(1.,YJ) + (1. - JET)*PSI0*TERM*EXARG0 +
     1  PSI1*SIN(YJ*ARG1)*EXARG1/ARG1)
      TERM = 0.
      IF (JET .NE. 0.0) TERM = JET * YJ / (1.0 + F)
      VUP(J) = -.5*(1. - TERM - PSI2*SIN(YJ*ARG2)*EXARG2/ARG2)
  740 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FINDSK(ISTART,IEND,J, ISK)
C                    SUBROUTINE LOCATES SHOCK WAVE ON LINE J BETWEEN
C                    ISTART AND IEND.  SHOCK IS LOCATED AT SHOCK POINT.
C                    IF NO SHOCK FOUND, ISK IS SET NEGATIVE.
C                    CALLED BY - CDCOLE.
      LOGICAL        PHYS
      INTEGER        PRTFLO   , SIMDEF
      COMMON /COM27/ CL       , DELTA   , DELRT2  , EMACH   , EMROOT  ,
     1               PHYS     , PRTFLO  , SIMDEF  , SONVEL  , VFACT   ,
     2               YFACT
      ISK = ISTART - 1
      U2 = PX(ISK,J)
    5 ISK = ISK + 1
      U1 = U2
      U2 = PX(ISK,J)
      IF(U1 .GT. SONVEL .AND. U2 .LE. SONVEL) GO TO 10
      IF (ISK .LT. IEND) GO TO 5
      ISK = - IEND
   10  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FIXPLT

C                         SETS UP ARRAYS FOR SUBROUTINE CPPLOT.
C                         CALLED BY - PRINT.

      COMMON          P(102,101),X(100)  , Y(100)
      COMMON / COM1/ IMIN     , IMAX    , IUP    , IDOWN   , ILE     ,
     1               ITE      , JMIN    , JMAX   , JUP     , JLOW    ,
     2               JTOP     , JBOT
      COMMON /COM13/ CDFACT   , CLFACT  , CMFACT , CPFACT  , CPSTAR
      COMMON /COM25/ CPL(100) , CPU(100)
```

157

```
      COMMON /COM38/ CPUP(101), CPLO(101) , CPS(101) , XP(101)          FIXPLT  11
                                                                        FIXPLT  12
      YMX = 3.*CPFACT                                                   FIXPLT  13
      YMN = -3.*CPFACT                                                  FIXPLT  14
      K = 0                                                             FIXPLT  15
      DO 150 I = IMIN,IMAX                                              FIXPLT  16
      K = K + 1                                                         FIXPLT  17
      GCP = -CPU(I)                                                     FIXPLT  18
      GCP = AMAX1(GCP,YMN)                                              FIXPLT  19
      GCP = AMIN1(GCP,YMX)                                              FIXPLT  20
      CPUP(K) = GCP                                                     FIXPLT  21
      GC1 = -CPL(I)                                                     FIXPLT  22
      GC1 = AMAX1(GC1,YMN)                                              FIXPLT  23
      GC1 = AMIN1(GC1,YMX)                                              FIXPLT  24
      CPLO(K) = GC1                                                     FIXPLT  25
      GC2 = -CPSTAR                                                     FIXPLT  26
      GC2 = AMAX1(GC2,YMN)                                              FIXPLT  27
      GC2 = AMIN1(GC2,YMX)                                              FIXPLT  28
      CPS(K) = GC2                                                      FIXPLT  29
      XP(K) = X(I)                                                      FIXPLT  30
  150 CONTINUE                                                          FIXPLT  31
      IMP = K + 1                                                       FIXPLT  32
      CPUP(IMP) = YMX                                                   FIXPLT  33
      CPLO(IMP) = YMN                                                   FIXPLT  34
      CPS(IMP)  = 0.0                                                   FIXPLT  35
      XP(IMP) = X(IMAX) + .001                                          FIXPLT  36
      CALL CPPLOT (XP, CPUP, CPLO, CPS, IMP)                            FIXPLT  37
      RETURN                                                            FIXPLT  38
      END                                                               FIXPLT  39
```

```
      SUBROUTINE GUESSP                                                 GUESSP   2
C              SUBROUTINE INITIALIZES P ARRAY AS FOLLOWS                GUESSP   3
C                   PSTART = 1  P SET TO ZERO                           GUESSP   4
C                   PSTART = 2  P READ FROM UNIT 7                      GUESSP   5
C                   PSTART = 3  P SET TO VALUES IN CORE                 GUESSP   6
C              IF PSTART = 2 OR 3, SOLUTION IS INTERPOLATED FROM        GUESSP   7
C              XOLD, YOLD TO X,Y                                        GUESSP   8
C              BOUNDARY CONDITIONS FOR P ON OUTER BOUNDARIES ARE        GUESSP   9
C              AUTOMATICALLY SET DURING INITIALIZATION                  GUESSP  10
C              CALLED BY - TSFOIL.                                      GUESSP  11
C                                                                       GUESSP  12
      COMMON          P(102,101),X(100) , Y(100)                        BLANK    2
      COMMON / COM1/ IMIN     , IMAX   , IUP    , IDOWN  , ILE    ,     COM1     2
     1               ITE      , JMIN   , JMAX   , JUP    , JLOW   ,     COM1     3
     2               JTOP     , JBOT                                    COM1     4
      COMMON / COM2/ AK       , ALPHA  , DUB    , GAM1   , RTK          COM2     2
      LOGICAL        ABORT                                              COM3     2
      COMMON / COM3/ IREF     , ABORT  , ICUT   , KSTEP                 COM3     3
      LOGICAL        AMESH                                              COM4     2
      COMMON / COM4/ XIN(100) , YIN(100), AMESH                         COM4     3
      INTEGER        PSTART                                             COM11    2
      LOGICAL        PSAVE                                              COM11    3
      COMMON /COM11/ ALPHA0   , CLOLD  , DELTA0 , DUB0   , EMACH0 ,     COM11    4
     1               IMINO    , IMAXO  , IMAXI  , JMINO  , JMAXO  ,     COM11    5
     2               JMAXI    , PSAVE  , PSTART , TITLE(20), TITLEO(20),COM11    6
     3               VOLD     , XOLD(100),YOLD(100)                     COM11    7
      COMMON /COM13/ CDFACT   , CLFACT , CMFACT , CPFACT , CPSTAR       COM13    2
      INTEGER        BCTYPE                                             COM20    2
      COMMON /COM20/ BCTYPE   , CIRCFF , FHINV  , POR    , CIRCTE       COM20    3
      COMMON /COM30/ PT(100)  , REST(804)                               GUESSP  21
C                                                                       GUESSP  22
C              BRANCH TO APPROPRIATE LOCATION                           GUESSP  23
      GO TO ( 200, 300, 400) , PSTART                                   GUESSP  24
  200 CONTINUE                                                          GUESSP  25
C              PSTART = 1                                               GUESSP  26
C              P SET TO ZERO                                            GUESSP  27
      DO 205 I=1,101                                                    GUESSP  28
      DO 205 J=1,102                                                    GUESSP  29
      P(J,I) = 0.                                                       GUESSP  30
  205 CONTINUE                                                          GUESSP  31
      DUB = 0.                                                          GUESSP  32
      CIRCFF = 0.                                                       GUESSP  33
      CIRCTE=0.0                                                        GUESSP  34
      RETURN                                                            GUESSP  35
  300 CONTINUE                                                          GUESSP  36
C              PSTART = 2                                               GUESSP  37
C              P, X, Y ARRAYS READ FROM UNIT 7 IN SUBROUTINE READIN     GUESSP  38
C              TOGETHER WITH INFORMATION ABOUT OLD SOLUTION             GUESSP  39
  400 CONTINUE                                                          GUESSP  40
C              PSTART = 3                                               GUESSP  41
C              ARRAYS FROM PREVIOUS CASE ARE ALREADY IN P,XOLD,YOLD     GUESSP  42
      DUB = DUB0                                                        GUESSP  43
      CIRCFF = CLOLD/CLFACT                                             GUESSP  44
      CIRCTE=CIRCFF                                                     GUESSP  45
C                                                                       GUESSP  46
C              FOR PSTART = 2 OR 3, OLD P ARRAY ON XOLD, YOLD MESH      GUESSP  47
C              MUST BE INTERPOLATED ONTO NEW X Y MESH.                  GUESSP  48
C                                                                       GUESSP  49
C              INTERPOLATE P FROM XOLD,YOLD TO X,YOLD                   GUESSP  50
C              CHECK TO SEE IF XOLD AND XIN ARE THE SAME MESH           GUESSP  51
      IF(IMAXI .NE. IMAXO) GO TO 450                                    GUESSP  52
      DO 410 I=IMIN, IMAXI                                              GUESSP  53
      TEST = ABS(XIN(I) - XOLD(I))                                      GUESSP  54
      IF(TEST .GT. .0001) GO TO 450                                     GUESSP  55
  410 CONTINUE                                                          GUESSP  56
C              XIN AND XOLD ARE SAME MESH.                              GUESSP  57
C              P ARRAY MAY BE INTERPOLATED BY SIMPLE DELETION OF        GUESSP  58
C              VALUES AT MESH POINTS DELETED IN SUBROUTINE CUTOUT       GUESSP  59
C              IF IREF .LE. ZERO, NO INTERPOLATION IS NEEDED            GUESSP  60
      IF(IREF .LE. 0) GO TO 500                                         GUESSP  61
      ISTEP = 2*IREF                                                    GUESSP  62
      DO 430 J = JMINO,JMAXO                                            GUESSP  63
      INEW = 0                                                          GUESSP  64
      DO 420 I = IMINO,IMAXO,ISTEP                                      GUESSP  65
      INEW = INEW + 1                                                   GUESSP  66
      P(J,INEW) = P(J,I)                                                GUESSP  67
  425 CONTINUE                                                          GUESSP  68
  430 CONTINUE                                                          GUESSP  69
C              INTERPOLATION IN X DIRECTION COMPLETE IF XIN AND         GUESSP  70
C              XOLD ARE THE SAME.                                       GUESSP  71
      GO TO 500                                                         GUESSP  72
  450 CONTINUE                                                          GUESSP  73
C              INTERPOLATE FROM XOLD TO X FOR ARBITRARY CASE            GUESSP  74
      DO 490 J = JMINO, JMAXO                                           GUESSP  75
      YP = YOLD(J)                                                      GUESSP  76
      DO 475 I = IMIN, IMAX                                             GUESSP  77
      XP = X(I)                                                         GUESSP  78
      IF(XP .LT. XOLD(IMINO)) GO TO 470                                 GUESSP  79
      IF(XP .GT. XOLD(IMAXO)) GO TO 470                                 GUESSP  80
C              NEW X MESH POINT WITHIN RANGE OF OLD X MESH              GUESSP  81
C              FIND VALUE OF XOLD .GT. XP                               GUESSP  82
      X2 = XOLD(1)                                                      GUESSP  83
      K = 1                                                             GUESSP  84
  455 K = K +1                                                          GUESSP  85
      X1 = X2                                                           GUESSP  86
      X2 = XOLD(K)                                                      GUESSP  87
      IF (X2 .LT. XP) GO TO 455                                         GUESSP  88
      IF (X2 .GT. XP) GO TO 460                                         GUESSP  89
      PT(I) = P(J,K)                                                    GUESSP  90
      GO TO 475                                                         GUESSP  91
  460 CONTINUE                                                          GUESSP  92
      P1 = P(J,K-1)                                                     GUESSP  93
      P2 = P(J,K)                                                       GUESSP  94
      PT(I) = P1 + (P2 - P1)/(X2 - X1) * (XP - X1)                      GUESSP  95
      GO TO 475                                                         GUESSP  96
  470 CONTINUE                                                          GUESSP  97
C              NEW X MESH POINT IS OUTSIDE RANGE OF OLD X MESH          GUESSP  98
C              FOR SUPERSONIC FREESTREAM SET P=0, FOR SUBSONIC          GUESSP  99
C              FREESTREAM, EXTRAPOLATE USING FAR FIELD SOLUTION         GUESSP 100
      PT(I) = 0.                                                        GUESSP 101
      IF(AK .GT. 0.) CALL EXTRAP(XP,YP,PT(I))                           GUESSP 102
  475 CONTINUE                                                          GUESSP 103
C              WRITE NEW VALUES FOR P INTO P ARRAY                      GUESSP 104
      DO 480 I = IMIN, IMAX                                             GUESSP 105
      P(J,I) = PT(I)                                                    GUESSP 106
  480 CONTINUE                                                          GUESSP 107
  490 CONTINUE                                                          GUESSP 108
  500 CONTINUE                                                          GUESSP 109
C              INTERPOLATE FROM X,YOLD TO X,Y                           GUESSP 110
C              CHECK TO SEE IF YIN AND YOLD ARE THE SAME MESH           GUESSP 111
C                                                                       GUESSP 112
      IF(JMAXI .NE. JMAXO) GO TO  550                                   GUESSP 113
      DO 510 J = JMIN, JMAXI                                            GUESSP 114
      TEST = ABS(YIN(J) - YOLD(J))                                      GUESSP 115
      IF(TEST .GT. .0001) GO TO 550                                     GUESSP 116
  510 CONTINUE                                                          GUESSP 117
C              YIN AND YOLD ARE THE SAME MESH                           GUESSP 118
C              P ARRAY MAY BE INTERPOLATED BY SIMPLE DELETION OF        GUESSP 119
C              VALUES AT MESH POINTS DELETED IN SUBROUTINE CUTOUT       GUESSP 120
C              IF IREF .LE. ZERO, NO INTERPOLATION IS NEEDED            GUESSP 121
      IF(IREF .LE. 0) GO TO 600                                         GUESSP 122
      JSTEP = 2*IREF                                                    GUESSP 123
      DO 530 I= IMIN, IMAX                                              GUESSP 124
      JNEW = 0
      DO 520 J = JMIND, JMAXD, JSTEP
      JNEW = JNEW + 1
      P(JNEW,I) = P(J,I)
  520 CONTINUE
  530 CONTINUE
C                   INTERPOLATION IN Y DIRECTION COMPLETE IF YIN AND
C                   YOLD ARE THE SAME.
      GO TO 600
  550 CONTINUE
C                   INTERPOLATE YOLD TO Y FOR ARBITRARY CASE
      DO 590 I = IMIN,IMAX
      XP = X(I)
      K = 2
      Y1 = YOLD(1)
      DO 575 J = JMIN,JMAX
      YP = Y(J)
      IF(YP .LT. YOLD(JMIND)) GO TO 570
      IF(YP .GT. YOLD(JMAXD)) GO TO 571
C                   NEW Y MESH POINT WITHIN RANGE OF OLD Y MESH
C                   FIND VALUE OF YOLD .GT. YP
      Y2 = Y1
      K = K - 1
  555 K = K + 1
      Y1 = Y2
      Y2 = YOLD(K)
      IF(Y2 .LE. YP) GO TO 555
      P1 = P(K-1,I)
      P2 = P(K,I)
      PT(J) = P1 + (P2 - P1) / (Y2 - Y1) * (YP - Y1)
      GO TO 575
C                   NEW Y MESH POINT OUTSIDE RANGE OF OLD Y MESH.
C                   SET P(J,I) = P(JMIND,I) OR, IF SUBSONIC FREESTREAM
C                   FREE AIRFLOW, EXTRAPOLATE FOR P(J,I) USING FARFLD
C                   FORMULA.
  570 CONTINUE
      PT(J) = P(JMIND,I)
      GO TO 572
  571 PT(J) = P(JMAXD,I)
  572 CONTINUE
      IF(AK .GT. 0. .AND. BCTYPE .EQ. 1) CALL EXTRAP(XP,YP,PT(J))
  575 CONTINUE
C                   PUT NEW P VALUES INTO P ARRAY
      DO 580 J = JMIN,JMAX
      P(J,I) = PT(J)
  580 CONTINUE
  590 CONTINUE
  600 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ISLIT ( X )
C                            ISLIT COMPUTES ILE AND ITE FOR ANY XMESH ARRAY.
C                            CALLED BY - CKMESH, CUTOUT, READIN, REFINE.
      COMMON / COM1/ IMIN     , IMAX    , IUP     , IDOWN   , ILE     ,
     1               ITE      , JMIN    , JMAX    , JUP     , JLOW    ,
     2               JTOP     , JBOT
      DIMENSION X(100)
      I = IMIN - 1
   10 CONTINUE
      I = I + 1
      IF (X(I) .LT. 0.0) GO TO 10
      ILE = I
   20 CONTINUE
      I = I + 1
      IF (X(I) .LE. 1.0) GO TO 20
      ITE = I - 1
      RETURN
      END
```

```
      SUBROUTINE JSLIT ( Y )
C                            JSLIT COMPUTES JLOW AND JUP FROM Y ARRAY.
C                            CALLED BY - CKMESH, CUTOUT, READIN, REFINE.
      COMMON / COM1/ IMIN     , IMAX    , IUP     , IDOWN   , ILE     ,
     1               ITE      , JMIN    , JMAX    , JUP     , JLOW    ,
     2               JTOP     , JBOT
      DIMENSION  Y(100)
      J = JMIN - 1
   10 CONTINUE
      J = J + 1
      IF (Y(J) .LT. 0.0) GO TO 10
      JLOW = J - 1
      JUP  = J
      RETURN
      END
```

```
      SUBROUTINE INPERR (I)
C                            FATAL ERROR IN INPUT. WRITE MESSAGE AND STOP.
C                            CALLED BY - READIN, SCALE.
      IF (I .EQ. 1) WRITE (6,901)
      IF (I .EQ. 2) WRITE (6,902)
      IF (I .EQ. 3) WRITE (6,903)
      IF (I .EQ. 4) WRITE (6,904)
      IF (I .EQ. 5) WRITE (6,905)
      IF (I .EQ. 6) WRITE (6,906)
      IF (I .EQ. 7) WRITE (6,907)
      STOP
  901 FORMAT(1H0,5X,45HIMAX OR JMAX IS GREATER THAN 100,NOT ALLOWED.)
  902 FORMAT(1H0,5X,30HX MESH POINTS NOT MONOTONIC INCREASING.)
  903 FORMAT(1H0,5X,30HY MESH POINTS NOT MONOTONIC INCREASING.)
  904 FORMAT(1H0,5X,44HMACH NUMBER NOT IN PERMITTED RANGE. (.5,2.0))
  905 FORMAT(1H0,5X,41HALPHA NOT IN PERMITTED RANGE. (-9.0, 9.0))
  906 FORMAT(1H0,5X,41HDELTA NOT IN PERMITTED RANGE. ( 0.0, 1.0))
  907 FORMAT(1H0,5X,45HAK=0. VALUE OF AK MUST BE INPUT SINCE PHYS=F.)
      END
```

```
      REAL FUNCTION LIFT (CLFACT)
C                 COMPUTE LIFT COEFFICIENT FROM JUMP IN P AT TRAILING
C                 EDGE OF AIRFOIL
C                 CALLED BY - PRINT1, SOLVE.
      COMMON          P(102,101),X(100)  , Y(100)
      COMMON / COM1/ IMIN     , IMAX    , IUP     , IDOWN   , ILE     ,
     1               ITE      , JMIN    , JMAX    , JUP     , JLOW    ,
     2               JTOP     , JBOT
      COMMON / COM7/ CJUP     , CJUP1   , CJLOW   , CJLOW1
      PTOP = CJUP*P(JUP,ITE) - CJUP1*P(JUP + 1 , ITE)
      PBOT = CJLOW*P(JLOW,ITE) - CJLOW1*P(JLOW-1,ITE)
      LIFT = 2.*CLFACT*(PTOP-PBOT)
      RETURN
      END
      SUBROUTINE MACHMP                                                 MACHMP   2
C                                                                       MACHMP   3
C                      SUBROUTINE TO PRINT MAP OF MACH NO. ROUNDED      MACHMP   4
C                      TO NEAREST .1. CALLED BY - PRINT.                MACHMP   5
C                                                                       MACHMP   6
      COMMON / COM1/ IMIN     , IMAX    , IUP     , IDOWN   , ILE     , COM1     2
     1               ITE      , JMIN    , JMAX    , JUP     , JLOW    , COM1     3
     2               JTOP     , JBOT                                    COM1     4
      COMMON /COM30/ MM(100) , REST(304)                                MACHMP   8
C                                                                       MACHMP   9
      INTEGER CHAR                                                      MACHMP  10
      DIMENSION CHAR(37)                                                MACHMP  11
      DATA CHAR/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1HA,1HB,1HC,1HD, MACHMP  12
     1 1HE,1HF,1HG,1HH,1HI,1HJ,1HK,1HL,1HM,1HN,1HO,1HP,1HQ,1HR,1HS,1HT, MACHMP  13
     2 1HU,1HV,1HW,1HX,1HY,1HZ,1H+/                                     MACHMP  14
      DATA IB/1H / , IP/1H+/ , IM/1H-/ , IL/1HL/ , IT/1HT/              MACHMP  15
C                                                                       MACHMP  16
      WRITE (6,400)                                                     MACHMP  17
      DO 60 K=2,JMAX                                                    MACHMP  18
      J         = JMAX - K + 2                                          MACHMP  19
      IJC = IB                                                          MACHMP  20
      IF ( J .EQ. JUP) IJC = IP                                         MACHMP  21
      IF ( J .EQ. JLOW ) IJC = IM                                       MACHMP  22
      DO 10 I=1,IMAX                                                    MACHMP  23
      MM(I)     = IB                                                    MACHMP  24
   10 CONTINUE                                                          MACHMP  25
      DO 50 I=2,IMAX                                                    MACHMP  26
      U         = PX(I,J)                                               MACHMP  27
      EM        = EMACH1 (U)                                            MACHMP  28
      IF (EM .GT. 0.0) GO TO 20                                         MACHMP  29
      MM(I) = CHAR(1)                                                   MACHMP  30
      GO TO 50                                                          MACHMP  31
   20 CONTINUE                                                          MACHMP  32
      KAR=INT(10.*EM+1.5)                                               MACHMP  33
      IF (KAR .GT. 37) KAR=37                                           MACHMP  34
      MM(I)=CHAR(KAR)                                                   MACHMP  35
   50 CONTINUE                                                          MACHMP  36
      WRITE (6,410) IJC, (MM(I),I=2,IMAX)                               MACHMP  37
   60 CONTINUE                                                          MACHMP  38
      DO 70 I=1,IMAX                                                    MACHMP  39
      MM(I) = IP                                                        MACHMP  40
      IF (I .EQ. ILE) MM(I) = IL                                        MACHMP  41
      IF (I .EQ. ITE) MM(I) = IT                                        MACHMP  42
   70 CONTINUE                                                          MACHMP  43
      WRITE (6,420) (MM(I),I=2,IMAX)                                    MACHMP  44
      RETURN                                                            MACHMP  45
                                                                        MACHMP  46
  400 FORMAT(1H1,50X,15HMACH NUMBER MAP/50X,15(1H*)///3X,4(6HSYMBOL,5X, MACHMP  47
     1 5HRANGE,14X)/5X,1H0,10X,9HM.LT.0.05,10X,20HA   0.95.LT.M.LE.1.05, MACHMP  48
     2 10X,20HK   1.95.LT.M.LE.2.05,10X,20HU   2.95.LT.M.LE.3.05/5X,    MACHMP  49
     3 20H1   0.05.LT.M.LE.0.15,10X,20HB   1.05.LT.M.LE.1.15,10X,       MACHMP  50
     4 20HL   2.05.LT.M.LE.2.15,10X,20HV   3.05.LT.M.LE.3.15/5X,        MACHMP  51
     5 20H2   0.15.LT.M.LE.0.25,10X,20HC   1.15.LT.M.LE.1.25,10X,       MACHMP  52
     6 20HM   2.15.LT.M.LE.2.25,10X,20HW   3.15.LT.M.LE.3.25/5X,        MACHMP  53
     7 20H3   0.25.LT.M.LE.0.35,10X,20HD   1.25.LT.M.LE.1.35,10X,       MACHMP  54
     8 20HN   2.25.LT.M.LE.2.35,10X,20HX   3.25.LT.M.LE.3.35/5X,        MACHMP  55
     9 20H4   0.35.LT.M.LE.0.45,10X,20HE   1.35.LT.M.LE.1.45,10X,       MACHMP  56
     A 20HO   2.35.LT.M.LE.2.45,10X,20HY   3.35.LT.M.LE.3.45/5X,        MACHMP  57
     B 20H5   0.45.LT.M.LE.0.55,10X,20HF   1.45.LT.M.LE.1.55,10X,       MACHMP  58
     C 20HP   2.45.LT.M.LE.2.55,10X,20HZ   3.45.LT.M.LE.3.55/5X,        MACHMP  59
     D 20H6   0.55.LT.M.LE.0.65,10X,20HG   1.55.LT.M.LE.1.65,10X,       MACHMP  60
     E 20HQ   2.55.LT.M.LE.2.65,10X,1H+,10X,9HM.GT.3.55/5X,             MACHMP  61
     F 20H7   0.65.LT.M.LE.0.75,10X,20HH   1.65.LT.M.LE.1.75,10X,       MACHMP  62
     G 20HR   2.65.LT.M.LE.2.75/5X,20H8   0.75.LT.M.LE.0.85,10X,        MACHMP  63
     H 20HI   1.75.LT.M.LE.1.85,10X,20HS   2.75.LT.M.LE.2.85/5X,        MACHMP  64
     I 20H9   0.85.LT.M.LE.0.95,10X,20HJ   1.85.LT.M.LE.1.95,10X,       MACHMP  65
     J 20HT   2.85.LT.M.LE.2.95///)                                     MACHMP  66
  410 FORMAT(11X,100A1)                                                 MACHMP  67
  420 FORMAT (12X,99A1)                                                 MACHMP  68
                                                                        MACHMP  69
      END                                                               MACHMP  70
```

```
      SUBROUTINE MLINE                                                  MLINE    2
C                  PRINTS COORDINATES WHERE SONIC VELOCITY IS COMPUTED  MLINE    3
C                  LINEAR INTERPOLATION BETWEEN MESH POINTS IS USED.    MLINE    4
C                  CALLED BY - PRINT.                                   MLINE    5
C **********************************************************************MLINE    6
                                                                        MLINE    7
      COMMON         P(102,101),X(100)  , Y(100)                        BLANK    2
      COMMON / COM1/ IMIN     , IMAX    , IUP     , IDOWN   , ILE     , COM1     2
     1               ITE      , JMIN    , JMAX    , JUP     , JLOW    , COM1     3
     2               JTOP     , JBOT                                    COM1     4
      COMMON / COM2/ AK       , ALPHA   , DUB     , GAM1    , RTK       COM2     2
      LOGICAL        PHYS                                               COM27    2
      INTEGER        PRTFLO   , SIMDEF                                  COM27    3
      COMMON /COM27/ CL       , DELTA   , DELRT2  , EMACH   , EMROOT  , COM27    4
     1               PHYS     , PRTFLO  , SIMDEF  , SONVEL  , VFACT   , COM27    5
     2               YFACT                                              COM27    6
      INTEGER        BCTYPE                                             COM28    2
      COMMON /COM28/ BCTYPE   , CIRCFF  , FHINV   , POR     , CIRCTE    COM28    3
      COMMON /COM30/ XSLPRT(200) , YSLPRT(200) , REST(4)                MLINE   13
                                                                        MLINE   14
      DIMENSION  XSONIC(10)                                             MLINE   15
      NPTS = 0                                                          MLINE   16
      KMIN = JMIN                                                       MLINE   17
      KMAX = JMAX                                                       MLINE   18
      JP = JMAX + JMIN                                                  MLINE   19
      DO 200 K=KMIN,KMAX                                                MLINE   20
      J = JP - K                                                        MLINE   21
      YPR = YFACT*Y(J)                                                  MLINE   22
      PX2 = PX(IMIN,J)                                                  MLINE   23
      M = 0                                                             MLINE   24
      IF (J .NE. JLOW) GO TO 170                                        MLINE   25
      IF (NPTS .NE. 0) WRITE (6,2070)                                   MLINE   26
  170 CONTINUE                                                          MLINE   27
      IMM = IMIN + 1                                                    MLINE   28
      DO 180 I=IMM,IMAX                                                 MLINE   29
      PX1 = PX2                                                         MLINE   30
      PX2 = PX(I,J)                                                     MLINE   31
      IF(PX1 .GT. SONVEL .AND. PX2 .GT. SONVEL) GO TO 180               MLINE   32
      IF(PX1 .LT. SONVEL .AND. PX2 .LT. SONVEL) GO TO 180               MLINE   33
      IF (NPTS .EQ. 0) WRITE (6,2000)                                   MLINE   34
      M = M+1                                                           MLINE   35
      RATIO = (SONVEL-PX1)/(PX2-PX1)                                    MLINE   36
      XSONIC(M) = X(I-1) + (X(I)-X(I-1))*RATIO                          MLINE   37
      NPTS = NPTS  + 1                                                  MLINE   38
      XSLPRT(NPTS) = XSONIC(M)                                          MLINE   39
      YSLPRT(NPTS) = YPR                                                MLINE   40
      IF(NPTS .GE. 200) GO TO 250                                       MLINE   41
  180 CONTINUE                                                          MLINE   42
      IF(M .EQ. 0) GO TO 200                                            MLINE   43
      WRITE(6,2040) YPR,(XSONIC(L),L=1,M)                               MLINE   44
  200  CONTINUE                                                         MLINE   45
       GO TO 300                                                        MLINE   46
  250  CONTINUE                                                         MLINE   47
       WRITE(6,2060)                                                    MLINE   48
       GO TO 400                                                        MLINE   49
  300  CONTINUE                                                         MLINE   50
       IF(NPTS .EQ.0) GO TO 400                                         MLINE   51
       YM = Y(JMIN)                                                     MLINE   52
       YX = Y(JMAX)                                                     MLINE   53
       DO 320 N=1,NPTS                                                  MLINE   54
       IF (YSLPRT(N) .NE. YM .AND. YSLPRT(N) .NE. YX) GO TO 320         MLINE   55
       IF ( AK .GT. 0.0 ) WRITE (6,2050)                                MLINE   56
       IF ( AK .LT. 0.0 .AND. BCTYPE .EQ. 1 ) WRITE (6,2050)            MLINE   57
  320  CONTINUE                                                         MLINE   58
       XMIN  = -.75                                                     MLINE   59
       XMAX  = 1.75                                                     MLINE   60
       XINCR =  .25                                                     MLINE   61
       YMIN  = -1.0                                                     MLINE   62
       YMAX  =  1.5                                                     MLINE   63
       YINCR =   .5                                                     MLINE   64
       CALL PLTSON(XSLPRT,YSLPRT,XMIN,XMAX,XINCR,YMIN,YMAX,YINCR,NPTS)  MLINE   65
  400  CONTINUE                                                         MLINE   66
       RETURN                                                           MLINE   67
                                                                        MLINE   68
 2000 FORMAT(1H1,50X,17HSONIC LINE OUTPUT/50X,17(1H*)/                  MLINE   69
     1 23H SONIC LINE COORDINATES//6X,1HY,10X,7HX-SONIC/)               MLINE   70
 2040 FORMAT(11F10.5)                                                   MLINE   71
 2050 FORMAT(20H0***** CAUTION *****/                                   MLINE   72
     *  34H SONIC LINE HAS REACHED A BOUNDARY/                          MLINE   73
     1 61H THIS VIOLATES ASSUMPTIONS USED TO DERIVE BOUNDARY CONDITIONS/ MLINE   74
     2 29H SOLUTION IS PROBABLY INVALID)                                MLINE   75
 2060 FORMAT(20H0***** CAUTION *****/36H NUMBER OF SONIC POINTS EXCEEDED MLINE   76
     1 200/25H ARRAY DIMENSION EXCEEDED/42H EXECUTION OF SUBROUTINE MLI MLINE   77
     2NE TERMINATED)                                                    MLINE   78
 2070 FORMAT(2X,13HBODY LOCATION)                                       MLINE   79
                                                                        MLINE   80
      END                                                               MLINE   81
      SUBROUTINE NEWISK(ISKOLD,J,ISKNEW)                                NEWISK   2
C                   FIND NEW LOCATION OF SHOCKWAVE (ISKNEW) ON LINE J   NEWISK   3
C                   GIVEN AN INITIAL GUESS FOR LOCATION (ISKOLD).       NEWISK   4
C                   SHOCK LOCATION IS DEFINED AS LOCATION OF SHOCK POINT. NEWISK 5
C                   IF NO SHOCK IS FOUND, ISKNEW IS SET NEGATIVE.       NEWISK   6
C                   CALLED BY - COCOL.                                  NEWISK   7
                                                                        NEWISK   8
      LOGICAL         PHYS                                              COM27    2
      INTEGER         PRTFLD   , SIMDEF                                 COM27    3
      COMMON /COM27/  CL       , DELTA   , DELRT2  , EMACH   , EMROOT  , COM27   4
     1                PHYS     , PRTFLD  , SIMDEF  , SONVEL  , VFACT   , COM27   5
     2                YFACT                                             COM27    6
      I2 = ISKOLD + 2                                                   NEWISK  10
      ISKNEW = ISKOLD - 3                                               NEWISK  11
      U2 = PX(ISKNEW,J)                                                 NEWISK  12
    5 ISKNEW = ISKNEW + 1                                               NEWISK  13
      U1 = U2                                                           NEWISK  14
      U2 = PX(ISKNEW,J)                                                 NEWISK  15
      IF(U1 .GT. SONVEL .AND. U2 .LE. SONVEL) GO TO 10                  NEWISK  16
      IF(ISKNEW .LT. I2) GO TO 5                                        NEWISK  17
C                   NO SHOCK POINT FOUND, TIP OF SHOCK REACHED.         NEWISK  18
      ISKNEW = - ISKNEW                                                 NEWISK  19
   10 RETURN                                                            NEWISK  20
      END                                                               NEWISK  21
```

```
      FUNCTION PITCH(CMFACT)                                            PITCH    2
C                   COMPUTE AIRFOIL PITCHING MOMENT ABOUT X = XM, Y=0   PITCH    3
C                   THE INTEGRAL OF(X-XM)(CPUPPER-CPLOWER) IS INTEGRATED PITCH   4
C                   BY PARTS TO GIVE AN INTEGRAL IN THE JUMP IN P PLUS A PITCH   5
C                   CONSTANT EQUAL TO THE JUMP IN P AT X = 1. TIMES THE PITCH    6
C                   MOMENT ARM AT THE TRAILING EDGE                     PITCH    7
C                   CALLED BY - PRINT1, SOLVE.                          PITCH    8
                                                                        PITCH    9
      COMMON          P(102,101),X(100)  , Y(100)                       BLANK    2
      COMMON / COM1/ IMIN     , IMAX    , IUP     , IDOWN   , ILE     , COM1     2
     1               ITE      , JMIN    , JMAX    , JUP     , JLOW    , COM1     3
     2               JTOP     , JBOT                                    COM1     4
      COMMON / COM7/ CJUP     , CJUP1   , CJLOW   , CJLOW1              COM7     2
      COMMON /COM30/ XI(100)  , ARG(100)  , REST(204)                   PITCH   13
                                                                        PITCH   14
C                   SET XM TO QUARTER CHORD                             PITCH   15
      XM = .25                                                          PITCH   16
      K = 0                                                             PITCH   17
      DO 10 I=ILE,ITE                                                   PITCH   18
      K = K + 1                                                         PITCH   19
      PTOP = CJUP*P(JUP,I) - CJUP1*P(JUP+1,I)                           PITCH   20
      PBOT = CJLOW*P(JLOW,I) - CJLOW1*P(JLOW-1,I)                       PITCH   21
      ARG(K) = PTOP - PBOT                                              PITCH   22
      XI(K) = X(I)                                                      PITCH   23
   10 CONTINUE                                                          PITCH   24
      CALL TRAP(XI,ARG,K,SUM)                                           PITCH   25
      PITCH = CMFACT*((1.-XM)*ARG(K) - SUM) * (-2.)                     PITCH   26
      RETURN                                                            PITCH   27
      END                                                               PITCH   28
```

```
      SUBROUTINE PLTSON (X,Y,XAXMIN,XMAX,XINCR,YAXMIN,YMAX,YINCR,NPTS)  PLTSON   2
                                                                        PLTSON   3
C                         THIS IS A MODIFIED VERSION OF PRNPLT AND      PLTSON   4
C                         XMAX, XINCR, YMAX, YINCR MUST BE SUPPLIED.    PLTSON   5
C                         XMIN, AND YMIN MUST ALSO BE SUPPLIED.         PLTSON   6
C                   PRINTER PLOT ROUTINE M. S. ITZKOWITZ  MAY,1967      PLTSON   7
C                   PLOTS THE (NPTS) POINTS GIVEN BY, X(I),Y(I)         PLTSON   8
C                   ON A 51 X 101 GRID USING A TOTAL OF 56 LINES ON     PLTSON   9
C                   THE PRINTER.                                        PLTSON  10
C                   IF EITHER INCREMENTAL STEP SIZE IS ZERO, THE        PLTSON  11
C                   PROGRAM EXITS.                                      PLTSON  12
C                   NEITHER OF THE INPUT ARRAYS ARE DESTROYED.          PLTSON  13
C                    CALLED BY - MSLINE.                                PLTSON  14
                                                                        PLTSON  15
      DIMENSION  X(NPTS),  Y(NPTS),  IGRID(105),  XAXIS(11)             PLTSON  16
      DIMENSION IBS(9)                                                  PLTSON  17
                                                                        PLTSON  18
      INTEGER  BLANK , DOT , STAR , DASH                                PLTSON  19
      DATA     BLANK , DOT , STAR , DASH / 1H , 1H., 1H*, 1H-/          PLTSON  20
      DATA      IBS/1HB,1HO,1HD,1HY,1H ,1HS,1HL,1HI,1HT/                PLTSON  21
                                                                        PLTSON  22
      WRITE(6,900)                                                      PLTSON  23
      WRITE(6,910)                                                      PLTSON  24
      IF (XINCR .EQ. 0. .OR. YINCR .EQ. 0.) GO TO 200                   PLTSON  25
      YRNGE  = YMAX - YAXMIN                                            PLTSON  26
      XRNGE  = XMAX - XAXMIN                                            PLTSON  27
      YVAL = YRNGE * .02                                                PLTSON  28
      XVAL = XRNGE * .01                                                PLTSON  29
      JSLT = IFIX( 51.*(YMAX/YRNGE) ) + 1                               PLTSON  30
      DXV = 1.0 / XVAL                                                  PLTSON  31
      DYV = 1.0 / YVAL                                                  PLTSON  32
      IZERO  = YMAX * DYV  + 1.5                                        PLTSON  33
      JZERO = 103.5 - XMAX * DXV                                        PLTSON  34
      IF (JZERO .GT. 103 .OR. JZERO .LT. 4) JZERO = 2                   PLTSON  35
      FIZERO = IZERO                                                    PLTSON  36
      FJZERO = JZERO                                                    PLTSON  37
      IGRID(1)      = BLANK                                             PLTSON  38
      IGRID(2)      = DOT                                               PLTSON  39
      IGRID(104)    = DOT                                               PLTSON  40
      IGRID(105)    = BLANK                                             PLTSON  41
      FIZER5 = FIZERO + .5                                              PLTSON  42
      FJZER5 = FJZERO + .5                                              PLTSON  43
C                         NOPW IS THE NUMBER OF PRINT WHEELS USED TO    PLTSON  44
C                                 SPAN THE BODY LENGTH.                 PLTSON  45
      NOPW = DXV                                                        PLTSON  46
C                         JS - THE NUMBER OF PRT WHEEL POSITIONS TO BE  PLTSON  47
C                              FILLED AT EACH END OF THE WORD *BODY SLIT* PLTSON 48
      JS = 0                                                            PLTSON  49
      IF (NOPW .GT. 9) JS = (NOPW-9) / 2                                PLTSON  50
                                                                        PLTSON  51
      WRITE (6,905)                                                     PLTSON  52
      WRITE (6,903)                                                     PLTSON  53
C                         LOOP TO SET UP ONE LINE EACH TIME THRU.       PLTSON  54
      DO 30 I=1,51                                                      PLTSON  55
C                         BLANK THE LINE TO BE PRINTED.                 PLTSON  56
      DO 2 J=3,103                                                      PLTSON  57
      IGRID(J) = BLANK                                                  PLTSON  58
    2 CONTINUE                                                          PLTSON  59
                                                                        PLTSON  60
C                         SEARCH ARRAY FOR POINTS ON THIS LINE.         PLTSON  61
                                                                        PLTSON  62
      DO 10 K=1,NPTS                                                    PLTSON  63
      ITEST = FIZER5 - Y(K) * DYV                                       PLTSON  64
      IF (ITEST .NE. I) GO TO 7                                         PLTSON  65
      J = FJZER5 + X(K) * DXV                                           PLTSON  66
      IF (J .GT. 103) J = 103                                           PLTSON  67
      IF (J .LT.   3) J = 3                                             PLTSON  68
      IGRID(J) = STAR                                                   PLTSON  69
    7 CONTINUE                                                          PLTSON  70
      IF (JSLT .EQ. I) GO TO 9                                          PLTSON  71
      GO TO 10                                                          PLTSON  72
    9 CONTINUE                                                          PLTSON  73
C                         WRITE *BODY SLIT* IF THIS IS THE JS LINE.     PLTSON  74
      J = JZERO + 1                                                     PLTSON  75
      IF (JS .EQ. 0) GO TO 4                                            PLTSON  76
      DO 3 L=1,JS                                                       PLTSON  77
      IF (IGRID(J) .NE. STAR) IGRID(J) = DASH                           PLTSON  78
      J = J + 1                                                         PLTSON  79
    3 CONTINUE                                                          PLTSON  80
    4 CONTINUE                                                          PLTSON  81
      DO 5 L=1,9                                                        PLTSON  82
      IF (IGRID(J) .NE. STAR) IGRID(J) = IBS(L)                         PLTSON  83
      J = J + 1                                                         PLTSON  84
    5 CONTINUE                                                          PLTSON  85
      IF (JS .EQ. 0) GO TO 10                                           PLTSON  86
      DO 6 L=1,JS                                                       PLTSON  87
      IF (IGRID(J) .NE. STAR) IGRID(J) = DASH                           PLTSON  88
      J = J + 1                                                         PLTSON  89
    6 CONTINUE                                                          PLTSON  90
   10 CONTINUE                                                          PLTSON  91
      IF (MOD(I,10) .EQ. 1) GO TO 13                                    PLTSON  92
      WRITE (6,901) IGRID                                               PLTSON  93
      GO TO 30                                                          PLTSON  94
   13 CONTINUE                                                          PLTSON  95
      FI = I - 1                                                        PLTSON  96
      YAXIS = YMAX - FI * YINCR * .1                                    PLTSON  97
      IF (ABS(YAXIS) .LT. YAXMIN) YAXIS = 0.                            PLTSON  98
      WRITE (6,902) YAXIS, (IGRID(J),J=1,105)                           PLTSON  99
   30 CONTINUE                                                          PLTSON 100
      WRITE (6,903)                                                     PLTSON 101
      WRITE (6,905)                                                     PLTSON 102
      DO 40 M=1,11                                                      PLTSON 103
      FM = I1 - M
      XAXIS(M) = XMAX - XINCR * FM
      IF (XAXIS(M) .LT. XAXMIN) XAXIS(M) = XAXMIN
   40 CONTINUE
      WRITE (6,904) XAXIS, NPTS
      RETURN

  500 WRITE (6,9900)
      CALL EXIT

  900 FORMAT(1H1,30X,17HSONIC LINE OUTPUT/1H0,17X(1H*)//)
  901 FORMAT(1H0,16A1)
  902 FORMAT(1X,F10.5,2X,1H+,105A1,1H+)
  903 FORMAT(15X,103(1H-))
  904 FORMAT(7X,11(F10.5),2X I,I4,5H PTS )
  905 FORMAT(16X,11(1H+,9X))
  913 FORMAT(15X,19HSONIC LINE PLOT,30X,8HY VS X,10X,
     1 10H * = SONIC POINT/)
 9900 FORMAT(44H0SCALING ERROR IN PAXPLT, EXECUTION TERMINATED )

      END
```

```
      SUBROUTINE PRBODY
C                 PRINTS OUT BODY GEOMETRY
C                 IF PHYS = .TRUE.  ALL DIMENSIONS ARE NORMALIZED BY
C                                   AIRFOIL CHORD
C                 IF PHYS = .FALSE. ALL DIMENSIONS EXCEPT X ARE
C                                   NORMALIZED BY CHORD LENGTH AND
C                                   THICKNESS RATIO.
C                 CALLED BY - BODY.

      COMMON / COM6/ FL(100)  , FXL(100), FU(100) , FXU(100),
     1               CAMBER(100), THICK(100),VOL   , XFOIL(100), IFOIL
      LOGICAL        PHYS
      INTEGER        PRTFLO   , SIMDEF
      COMMON /COM27/ CL       , DELTA   , DELRT2 , EMACH   , EMROOT  ,
     1               PHYS     , PRTFLO  , SIMDEF , SONVEL  , VFACT   ,
     2               YFACT

      WRITE(6,1000)
C                 FIND MAXIMUM THICKNESS AND CAMBER
      THMAX = 0.
      CAMAX = 0.
      DO 10 I = 1,IFOIL
      THMAX = AMAX1(THMAX,THICK(I))
      CAMAX = AMAX1(CAMAX,CAMBER(I))
   10 CONTINUE
      THMAX = 2.*THMAX
      IF(PHYS) GO TO 100
C                 PRINTOUT IN SIMILARITY VARIABLES
      WRITE(6,1001) THMAX
      WRITE(6,1002) VOL,CAMAX
      WRITE(6,1003)
      DO 40 I=1,IFOIL
      WRITE (6,1005) XFOIL(I),FU(I),FXU(I),FL(I),FXL(I),THICK(I),
     1               CAMBER(I)
   40 CONTINUE
      RETURN
  100 CONTINUE
C                 PRINTOUT IN PHYSICAL VARIALBES
      THMAX = DELTA*THMAX
      CAMAX = DELTA*CAMAX
      WRITE(6,1006) THMAX
      VOLUME = VOL*DELTA
      WRITE(6,1002) VOLUME,CAMAX
      WRITE(6,1004)
      DO 110 I=1,IFOIL
      YUP = DELTA*FU(I)
      YXUP = DELTA*FXU(I)
      YLO = DELTA*FL(I)
      YXLO = DELTA*FXL(I)
      TH = DELTA*THICK(I)
      CA = DELTA*CAMBER(I)
      WRITE(6,1005) (XFOIL(I),YUP,YXUP,YLO,YXLO,TH,CA)
  110 CONTINUE
      RETURN
 1000 FORMAT(1H1,35X,23HAIRFOIL GEOMETRY OUTPUT/36X,23(1H*)///)
 1001 FORMAT(33H PRINTOUT IN SIMILARITY VARIABLES,
     *  40X, 15HMAX THICKNESS =,F12.8)
 1002 FORMAT(16H AIRFOIL VOLUME =,F12.8,30X,15HMAX CAMBER    =,F12.8//
     1 2X,11HINPUT GRID,16X,14HUPPER  SURFACE,17X,14HLOWER  SURFACE)
 1003 FORMAT(6X,1HI,1X,2(15X,1HF,9X,5HDF/DX),11X,9HTHICKNESS,4X,
     1      6HCAMBER/)
 1004 FORMAT(6X,1HX,1X,2(15X,1HY,9X,5HDY/DX),11X,9HTHICKNESS,4X,
     1      6HCAMBER/)
 1005 FORMAT(1X,F12.8,5X,2F12.8,3X,2F12.8,2F12.8))
 1006 FORMAT(58H PRINTOUT IN PHYSICAL VARIABLES NORMALIZED BY CHORD LENG
     1TH, 23X, 15HMAX THICKNESS =,F12.8)
      END
```

```
      SUBROUTINE PRINT

C                         SUBROUTINE FOR MAIN OUTPUT PRINT CONTROL.
C                         PRINTS RELATIVE PARAMETERS AND CALLS
C                         SPECIALIZED PRINT/PLOT SUBROUTINES AS REQUIRED.

C                              PRINTOUT DESCRIPTION
C                  IF PHYS IS .TRUE. THE FOLLOWING INFORMATION IS
C                  PRINTED IN PHYSICALLY SCALED VARIABLES (I.E. USUAL
C                  AERODYNAMIC TERMS)
C                      X,Y ARE COORDINATES NON-DIMENSIONALIZED BY
C                          AIRFOIL CHORD.
C                      CP = (P - PINF)/(QINF) = -2.*(DPP/DX)
C                      CD = DRAG COEFF = D/(QINF)*AIRFOIL CHORD
C                      CPSTAR = CRITICAL PRESSURE COEFFICIENT
C                  IF PHYS IS .FALSE. THE ABOVE INFORMATION IS PRINTED
C                  IN TRANSONIC SIMILARITY VARIABLES.
C
C                  PRINTOUT OF SONIC LINE ORDINATES IN SUBR. MILINE
C                  INCLUDING PRINTPLOT OF SONIC LINE ORDINATES
C                  CALLED BY - TSFOIL.
C ****************************************************************************

      COMMON / COM2/ AK       , ALPHA   , DUB     , GAM1    , RTK
      LOGICAL        ABORT
      COMMON / COM3/ IREF     , ABORT   , ICUT    , KSTEP
      COMMON / COM6/ FL(100)  , FXL(100), FU(100) , FXU(100),
     1            CAMBER(100), THICK(100),VOL     , XFOIL(100), IFOIL
      INTEGER        PSTART
      LOGICAL        PSAVE
      COMMON /COM11/ ALPHAO   , CLOLD   , DELTAO  , DUBO    , EMACHO  ,
     1               IMINO    , IMAXO   , IMAXI   , JMINO   , JMAXO   ,
     2               JMAXI    , PSAVE   , PSTART  , TITLE(20), TITLEO(20),
     3               VOLO     , XOLD(100),YOLD(100)
      COMMON /COM13/ CDFACT   , CLFACT  , CMFACT  , CPFACT  , CPSTAR
      LOGICAL        FCR      , KUTTA
      COMMON /COM14/ CLSET    , FCR     , KUTTA   , WCIRC
      LOGICAL        PHYS
      INTEGER        PRTFLO   , SIMDEF
      COMMON /COM27/ CL       , DELTA   , DELRT2  , EMACH   , EMROOT  ,
     1               PHYS     , PRTFLO  , SIMDEF  , SONVEL  , VFACT   ,
     2               YFACT
      INTEGER        BCTYPE
      COMMON /COM28/ BCTYPE   , CIRCFF  , FHINV   , POR     , CIRCTE

      DIMENSION  TPH(6)  , SIM(8) , BCT(15) , FCP(14)

      DATA  TPH /4HSIMI,4HLARI,2HTY,4H PHY,4HSICA,2HL /
      DATA  SIM /4HCOLE,4H    ,4HSPRE,4HITER,4HKRUP,4HP    ,
     1           4HUSER,4H     /
      DATA  BCT /4HFREE,4H AIR,4H    ,4HSOLI,4HD WA,4HLL  ,4HFREE,
     1     4H JET,4H    ,4HSLOT,4HTED ,4HWALL,4HPORO,4HUS W,4HALL /
      DATA  FCP /4HFULL,4HY CO,4HNSER,4HVATI,4HVE, ,4H    ,4H    ,
     1           4HNOT ,4HCONS,4HERVA,4HTIVE,4H AT ,4HSHOC,4HK.   /

      WRITE (6,900)
      IS = 1
      IF (PHYS) IS = 4
      IE = IS + 2
      WRITE (6,901) (TPH(I),I=IS,IE)
      IE = 2 * SIMDEF
      IS = IE - 1
      WRITE (6,902) (SIM(I),I=IS,IE)
      IE = 3 * BCTYPE
      IS = IE - 2
      WRITE (6,903) (BCT(I),I=IS,IE)
      IS = 8
      IF (FCR) IS = 1
      IE = IS + K
      WRITE (6,404) (FCP(I),I=IS,IE)
      IF (KUTTA) GO TO 10
      WRITE (6,405)
      GO TO 20
   10 CONTINUE
      WRITE (6,406)
   20 CONTINUE
      ALPHVF = ALPHA * VFACT
      WRITE (6,411)
      IF ( PHYS ) WRITE (6,409) EMACH, DELTA
      WRITE (6,407) ALPHVF, AK
      DUBDEL=DUB*DELTA
      VOLDEL=VOL*DELTA
      IF (AK .GT. 0.0) WRITE(6,408) DUBDEL,VOLDEL
      IF ( PHYS ) WRITE(6,410)CPFACT,CDFACT,CMFACT,CLFACT,YFACT,VFACT
      CALL PRINT1
      CALL PRTMC
      CALL MACHMP
      IF ( ABORT ) GO TO 60
      CALL FIXPLT
      IF (BCTYPE .EQ. 1) GO TO 40
      IF (BCTYPE .EQ. 3) GO TO 40
      CALL PRTWAL
   40 CONTINUE
      CALL MILINE
      IF (PRTFLO .EQ. 1) GO TO 50
      CALL PRTFLD
   50 CONTINUE
      CALL CDCOLE
   60 CONTINUE
      RETURN
C
  400 FORMAT(1H1,5X,18(1H*)/5X,1H*,16X,1H*/5X,18H*  FINAL OUTPUT  */
     1 5X,1H*,16X,1H*/5X,18(1H*))
  401 FORMAT(14H0 PRINTOUT IN ,2A4,A2,11H VARIABLES.)
  402 FORMAT(41H0 DEFINITION OF SIMILARITY PARAMETERS BY ,2A4)
  403 FORMAT(25H0 BOUNDARY CONDITION FOR ,3A4)
  404 FORMAT(27H0 DIFFERENCE EQUATIONS ARE ,7A4)
  405 FORMAT(37H0 LIFT COEFFICIENT SPECIFIED BY USER.)
  406 FORMAT(30H0 KUTTA CONDITION IS ENFORCED.)
  407 FORMAT(13X,7HALPHA =,F12.7/17X,3HK =,F12.7)
  408 FORMAT(2X,18HDOUBLET STRENGTH =,F12.7/4X,16HAIRFOIL VOLUME =,
     1 F12.7)
  409 FORMAT(14X,6HMACH =,F12.7/13X,7HDELTA =,F12.7)
  410 FORMAT(1H0,6X,38HPARAMETERS USED TO TRANSFORM VARIABLES/
     1     18X,20HTO TRANSONIC SCALING//12X,8HCPFACT =F12.7/
     2     12X,8HCDFACT =,F12.7/12X,8HCMFACT =,F12.7/12X,8HCLFACT =,F12.7
     3    /13X,7HYFACT =,F12.7/ 13X,7HVFACT =F12.7)
  411 FORMAT(1H0)

      END




      SUBROUTINE PRINT1
C
C                    PRINTS PRESSURE COEFFICIENT AND MACH NUMBER
C                    ON Y=0 LINE, AND PLOTS CP ALONG SIDE OF PRINT.
C                    CALLED BY - TSFOIL, PRINT.

      COMMON         P(102,101),X(100) , Y(100)
      COMMON / COM1/ IMIN     , IMAX    , IUP     , IDOWN   , ILE     ,
     1               ITE      , JMIN    , JMAX    , JUP     , JLOW    ,
     2               JTOP     , JBOT
      LOGICAL        ABORT
      COMMON / COM3/ IREF     , ABORT   , ICUT    , KSTEP
      COMMON / COM7/ CJUP     , CJUP1   , CJLOW   , CJLOW1
      COMMON /COM13/ CDFACT   , CLFACT  , CMFACT  , CPFACT  , CPSTAR
      COMMON /COM25/ CPL(100) , CPU(100)
      LOGICAL        PHYS
      INTEGER        PRTFLO   , SIMDEF
      COMMON /COM27/ CL       , DELTA   , DELRT2  , EMACH   , EMROOT  ,
     1               PHYS     , PRTFLO  , SIMDEF  , SONVEL  , VFACT   ,
     2               YFACT
      COMMON /COM30/ EM1L(100), EM1U(100), YR(100), REST(204)

      REAL LIFT
      DIMENSION  LINE1(60)  , TMAC(2)

      DATA  TMAC / 2HM1, 2HK1/ , IB /1H /, IL / 1HL/,IU /1HU/, IS/1H*/
      DATA  IBB/ 1HB/

      CL = LIFT (CLFACT)
      CM = PITCH (CMFACT)
      CPMIN =  1.0E37
      CPMAX = -CPMIN
      IEM = 0
      CJ01 = -Y(JLOW)/(Y(JUP)-Y(JLOW))
      CJ02 = Y(JUP)/(Y(JUP)-Y(JLOW))

      DO 10 I=IMIN,IMAX
      UL      = CJLOW*PX(I,JLOW) - CJLOW1*PX(I,JLOW-1)
      IF(I .GT. ITE)  UL=CJ01*PX(I,JUP) + CJ02*PX(I,JLOW)
      IF(I .LT. ILE)  UL=CJ01*PX(I,JUP) + CJ02*PX(I,JLOW)
      CPL(I)  = -2.0 * UL * CPFACT
      EM1L(I) = EMACH1(UL)
      IF (EM1L(I) .GT. 1.3) IEM = 1
      UU      = CJUP *PX(I,JUP)  - CJUP1 *PX(I,JUP+1)
      IF (I .GT. ITE) UU = UL
      IF (I .LT. ILE) UU = UL
      CPU(I)  = -2.0 * UU * CPFACT
      EM1U(I) = EMACH1(UU)
      IF (EM1U(I) .GT. 1.3) IEM = 1
      CPMAX   = AMAX1(CPMAX,CPU(I), CPL(I))
      CPMIN   = AMIN1(CPMIN,CPU(I), CPL(I))
   10 CONTINUE

      CPLARG  = AMAX1(CPMAX,ABS(CPMIN))
      UNPCOL  = CPLARG / 29.

C                            LOCATE CP* FOR PRINTER PLOT

      COL           = -CPSTAR / UNPCOL
      NCOL          = SIGN((ABS(COL)+.5), COL)
      NCOLS         = NCOL + 30

C                            PRINT SINGLE VARIABLES

      IF (IREF .EQ. 2) WRITE (6,270)
      IF (IREF .EQ. 1) WRITE (6,280)
      IF (IREF .EQ. 0) WRITE (6,290)
      WRITE (6,260)
      WRITE (6,200) CL, CM, CPSTAR
      IF (CPL(IMIN) .LT. CPSTAR .AND. CPL(IMIN+1) .GT. CPSTAR) GO TO 70
      WRITE (6,210)
      KT = 2
      IF ( PHYS ) KT = 1

C                            PRINT COLUMN HEADERS.

      WRITE (6,220) TMAC(KT) , TMAC(KT)

      DO 40 I=IMIN,IMAX

      DO 30 K=1,60
      LINE1(K) = IB
   30 CONTINUE

      COL           = -CPU(I) / UNPCOL
      NCOL          =  SIGN((ABS(COL) + .5) , COL)
      NCOLU         =  NCOL + 30
      LINE1(NCOLU)  =  IU

      COL           = -CPL(I) / UNPCOL
      NCOL          =  SIGN((ABS(COL) + .5) , COL)
      NCOLL         =  NCOL + 30
      LINE1(NCOLL)  =  IL
      IF (NCOLL .EQ. NCOLU) LINE1(NCOLL) = IBB
      IF (IABS(NCOLS) .LT. 61) LINE1(NCOLS) = IS

      IF (I .EQ. ILE) WRITE (6,230)
      WRITE (6,250) I, X(I), CPL(I), EM1L(I), CPU(I), EM1U(I), LINE1
      IF (I .EQ. ITE) WRITE (6,240)

   40 CONTINUE
      IF ( IEM .EQ. 0 ) GO TO 50
      IF ( PHYS ) WRITE (6,300)
   50 CONTINUE
      DO 60 J=JMIN,JMAX
      YR(J) = Y(J) * YFACT
   60 CONTINUE
      WRITE (6,310) JMIN,JMAX,(YR(J),J=JMIN,JMAX)
      RETURN
   70 CONTINUE
      WRITE (6,320)
      ABORT = .TRUE.
      RETURN
```

Microfilmed From
Best Available Copy

```
200 FORMAT(45X,28HB = BEFORE OR BEHIND AIRFOIL/10X,4HCL =,F10.6,61X,    PRINT1 109
   1 25HU = UPPER AIRFOIL SURFACE/10X,4HCM =,F10.6,61X,                  PRINT1 110
   2 25HL = LOWER AIRFOIL SURFACE/9X,5HCP* =,F10.6,61X,                  PRINT1 111
   3 21H* = CRITICAL PRESSURE)                                           PRINT1 112
210 FORMAT(1H0,27X,5HLOWER,23X,5HUPPER/20X,4HY=0-,24X,4HY=0+)           PRINT1 113
220 FORMAT(3X,1HI,4X,1HX,10X,2HCP,10X,A2,14X,2HCP,10X,A2/)              PRINT1 114
230 FORMAT(25X,20HAIRFOIL LEADING EDGE,40X,20HAIRFOIL LEADING EDGE)     PRINT1 115
240 FORMAT(25X,21HAIRFOIL TRAILING EDGE,44X,21HAIRFOIL TRAILING EDGE)   PRINT1 116
250 FORMAT(1H ,I3,3F12.6,4X,2F12.6,2X,A5A1)                             PRINT1 117
260 FORMAT(60H  FORCE COEFFICIENTS, PRESSURE COEFFICIENT, AND MACH NUM  PRINT1 118
   1BER /20X,59HFOR SIMILARITY PARAMETERS ON BODY AND DIVIDING STREAM L PRINT1 119
   2INE.)                                                                PRINT1 120
270 FORMAT(1H1,52X,28HFINAL OUTPUT FOR COARSE MESH//)                   PRINT1 121
280 FORMAT(1H1,50X,28HFINAL OUTPUT FOR MEDIUM MESH//)                   PRINT1 122
290 FORMAT(1H1,53X,26HFINAL OUTPUT FOR FINE MESH//)                     PRINT1 123
300 FORMAT(20H0***** CAUTION *****/                                      PRINT1 124
   1 32H MAXIMUM MACH NUMBER EXCEEDS 1.3/                                PRINT1 125
   2 69H SHOCK JUMPS IN ERROR IF UPSTREAM NORMAL MACH NUMBER GREATER T  PRINT1 126
   3HAN 1.3)                                                             PRINT1 127
310 FORMAT(///1X,A4,10H-GRID VALUES,1X,5HPHI(J) J=,I4,3H TO,I4/          PRINT1 128
   1 (1X,9F12.6))                                                        PRINT1 129
320 FORMAT(1H0,///,60H DETACHED SHOCK WAVE UPSTREAM OF X-MESH,SOLUTION  PRINT1 130
   1TERMINATED.)                                                         PRINT1 131
                                                                         PRINT1 132
    END                                                                  PRINT1 133
```

```
      SUBROUTINE PRTFLD                                                 PRTFLD   2
                                                                        PRTFLD   3
C                          PRINTS PRESSURE COEFFICIENT, FLOW ANGLE AND  PRTFLD   4
C                          MACH NUMBER IN FLOW FIELD. NUMBER OF J LINES PRTFLD   5
C                          PRINTED IS DETERMINED FROM THE INPUT VALUE OF PRTFLD   6
C                          PRTFLD.                                      PRTFLD   7
C                              PRTFLD = 1 , NONE.                       PRTFLD   8
C                              PRTFLD = 2 , ALL J LINES EXCEPT JB.      PRTFLD   9
C                              PRTFLD = 3 , THREE J LINES AROUND JERROR. PRTFLD  10
C                         CALLED BY - PRINT.                            PRTFLD  11
                                                                        PRTFLD  12
      COMMON          P(102,101),X(100) , Y(100)                        BLANK    2
      COMMON / COM1/ IMIN    , IMAX    , IUP     , IDOWN   , ILE     ,  COM1     2
     1               ITE     , JMIN    , JMAX    , JUP     , JLOW    ,  COM1     3
     2               JTOP    , JBOT                                     COM1     4
      COMMON /COM13/ CDFACT  , CLFACT  , CMFACT  , CPFACT  , CPSTAR     COM13    2
      LOGICAL        OUTERR                                             COM18    2
      COMMON /COM18/ ERROR   , I1      , I2      , IERROR  , JERROR  ,  COM18    3
     1               OUTERR  , EMU(100,2)          , VC(100) ,          COM18    4
     2               WI      , BCIRC   , POLD(100,2)                    COM18    5
      LOGICAL        PHYS                                               COM27    2
      INTEGER        PRTFLD  , SIMDEF                                   COM27    3
      COMMON /COM27/ CL      , DELTA   , DELRT2  , EMACH   , EMROOT  ,  COM27    4
     1               PHYS    , PRTFLD  , SIMDEF  , SONVEL  , VFACT   ,  COM27    5
     2               YFACT                                              COM27    6
      COMMON /COM30/ JLIN(100), JEST(304)                               PRTFLD  18
                                                                        PRTFLD  19
      DIMENSION  TRAC(2), CPPR(3), PYPR(3), YPRINT(3), PRT(10), EMLC(3) PRTFLD  20
      DATA   TRAC / 2HCL, 2HCL/                                         PRTFLD  21
      DATA   PRT  / 4HMACH,4H NUM,4HBER),4H    ,4H    ,                 PRTFLD  22
     1              4H(SIM),4HILAR),4HITY P,4HARAM,4HETER)/             PRTFLD  23
      IF (PRTFLD.EQ. 2) GO TO 25                                        PRTFLD  24
C                          LOCATE LINES AROUND JERROR.                  PRTFLD  25
      JL = 3                                                            PRTFLD  26
      IF (JERROR .EQ. JMIN) GO TO 10                                    PRTFLD  27
      IF (JERROR .EQ. JLOW) GO TO 20                                    PRTFLD  28
      IF (JERROR .EQ. JUP ) GO TO 10                                    PRTFLD  29
      IF (JERROR .EQ. JMAX) GO TO 20                                    PRTFLD  30
      JLIN(1) = JERROR - 1                                              PRTFLD  31
      JLIN(2) = JERROR                                                  PRTFLD  32
      JLIN(3) = JERROR + 1                                              PRTFLD  33
      GO TO 30                                                          PRTFLD  34
   10 CONTINUE                                                          PRTFLD  35
      JLIN(1) = JERROR                                                  PRTFLD  36
      JLIN(2) = JERROR + 1                                              PRTFLD  37
      JLIN(3) = JERROR + 2                                              PRTFLD  38
      GO TO 30                                                          PRTFLD  39
   20 CONTINUE                                                          PRTFLD  40
      JLIN(1) = JERROR - 2                                              PRTFLD  41
      JLIN(2) = JERROR - 1                                              PRTFLD  42
      JLIN(3) = JERROR                                                  PRTFLD  43
      GO TO 30                                                          PRTFLD  44
   25 CONTINUE                                                          PRTFLD  45
      JL = JMAX - JMIN + 1                                              PRTFLD  46
C                          FILL JLIN ARRAY WITH VALUES OF J.            PRTFLD  47
      K = 1                                                             PRTFLD  48
      DO 28 J=JMIN,JMAX                                                 PRTFLD  49
      JLIN(K) = J                                                       PRTFLD  50
      K = K + 1                                                         PRTFLD  51
   28 CONTINUE                                                          PRTFLD  52
   30 CONTINUE                                                          PRTFLD  53
C                                                                       PRTFLD  54
C                          PRINT FLOW FIELD IN 3 J LINES PER PAGE.      PRTFLD  55
      DO 100 MPR = 1,JL,3                                               PRTFLD  56
      MPREND = MIN0(MPR+2,JL)                                           PRTFLD  57
      DO  40 M=MPR,MPREND                                               PRTFLD  58
      MQ = M-MPR+1                                                      PRTFLD  59
      JQ = JLIN(M)                                                      PRTFLD  60
      YPRINT(MQ) = Y(JQ)*YFACT                                          PRTFLD  61
   40 CONTINUE                                                          PRTFLD  62
C                          WRITE PAGE HEADER.                           PRTFLD  63
      I1 = 1                                                            PRTFLD  64
      IF ( PHYS ) I1 = 6                                                PRTFLD  65
      I2 = I1 + 4                                                       PRTFLD  66
      WRITE (6,900) (PRT(I),I=I1,I2) , CPSTAR                           PRTFLD  67
      WRITE(6, 901) (JLIN(M),M=MPR,MPREND)                              PRTFLD  68
      WRITE(6, 902)(YPRINT(M),M=1,MQ)                                   PRTFLD  69
      KT = 2                                                            PRTFLD  70
      IF ( PHYS ) KT = 1                                                PRTFLD  71
      WRITE (6,903) TRAC(KT), TRAC(KT), TRAC(KT)                        PRTFLD  72
      DO  60 I=IMIN,IMAX                                                PRTFLD  73
      DO  50 M=MPR,MPREND                                               PRTFLD  74
      MQ = M-MPR+1                                                      PRTFLD  75
      J = JLIN(M)                                                       PRTFLD  76
      U = PX(I,J)                                                       PRTFLD  77
      CPPR(MQ) = -2.0 * CPFACT * U                                      PRTFLD  78
      PYPR(MQ) =   VFACT*PY(I,J)                                        PRTFLD  79
      EMLC(MQ)  = EMACH1(U)                                             PRTFLD  80
   50 CONTINUE                                                          PRTFLD  81
      WRITE (6,904) I,X(I),((CPPR(M),PYPR(M),EMLC(M)),M=1,MQ)           PRTFLD  82
   60 CONTINUE                                                          PRTFLD  83
  100 CONTINUE                                                          PRTFLD  84
      RETURN                                                            PRTFLD  85
                                                                        PRTFLD  86
  900 FORMAT(1H1,47HPRESSURE COEFFICIENTS, FLOW ANGLES, AND LOCAL ,5A4/ PRTFLD  87
     1        28H ON Y=CONSTANT LINES/9H CPSTAR =,F12.7//)              PRTFLD  88
  901 FORMAT(15X,3(11X,2HJ=,I4,15X))                                    PRTFLD  89
  902 FORMAT(15X,3(12X,2HY=,F13.6,12X))                                 PRTFLD  90
  903 FORMAT(4H0   I,8X,1HX,3(6X,2HCP,8X,5HTHETA,7X,A2,6X)//)           PRTFLD  91
  904 FORMAT(1X,I3,2X,F10.6,1X,3(2X,3F11.6,1X))                         PRTFLD  92
                                                                        PRTFLD  93
      END                                                               PRTFLD  94
```

```
      SUBROUTINE PRTMC                                                  PRTMC    2
C                          SUBROUTINE TO PRINT A CHARACTER FOR EACH     PRTMC    3
C                          POINT IN THE GRID DESCRIBING THE TYPE OF     PRTMC    4
C                          FLOW.    S , SHOCK POINT                     PRTMC    5
C                                   H , HYPERBOLIC POINT                PRTMC    6
C                                   P , PARABOLIC POINT                 PRTMC    7
C                                   - , ELLIPTIC POINT                  PRTMC    8
C                          CALLED BY - PRINT.                           PRTMC    9
                                                                        PRTMC   10
      COMMON          P(102,101),X(100) , Y(100)                        BLANK    2
      COMMON / COM1/ IMIN    , IMAX    , IUP     , IDOWN   , ILE     ,  COM1     2
     1               ITE     , JMIN    , JMAX    , JUP     , JLOW    ,  COM1     3
     2               JTOP    , JBOT                                     COM1     4
      COMMON / COM2/ AK      , ALPHA   , DUB     , GAM1    , RTK       COM2     2
      COMMON /COM22/ CXC(100) , CXL(100), CXR(100), CXXC(100),CXXL(100), COM22    2
     1               CXXR(100), C1(100)                                 COM22    3
                                                                        PRTMC   15
      COMMON  /COM30/ IPC(100) , VT(100,2), REST(204)                   PRTMC   16
      DATA  IHP/1HP/, IHH/1HH/, IHS/1HS/, IHD/1H-/, IB/1H /             PRTMC   17
                                                                        PRTMC   18
      WRITE (6,100)                                                     PRTMC   19
      DO 5 I=1,99,2                                                     PRTMC   20
      IPC(I)   = IS                                                     PRTMC   21
      IPC(I+1) = IB                                                     PRTMC   22
    5 CONTINUE                                                          PRTMC   23
      DO 10 J=JMIN,JMAX                                                 PRTMC   24
      VT(J,1) = C1(2)                                                   PRTMC   25
   10 CONTINUE                                                          PRTMC   26
```

164

```
      DO 60 K=JMIN,JMAX
      J = JMAX - K + 1
      DO 50 I=IUP,IDOWN
      VT(1,I) = VT(2,I)
      VT(2,I) = C(I)-(CDL(I)*PHI(J,I-1)+CDC(I)*PHI(J,I)+CDR(I)*PHI(J,I+1))
      IF (VT(1,I) .LT. 0.0) GO TO 30
      IF (VT(2,I) .LT. 0.0) GO TO 20
C                         PARABOLIC POINT   (SONIC)
      IPC(I) = IHP
      GO TO 50
   20 CONTINUE
C                         HYPERBOLIC POINT   (SUPERSONIC)
      IPC(I) = IHH
      GO TO 50
   30 CONTINUE
      IF (VT(2,I) .LT. 0.0) GO TO 40
C                         ELLIPTIC POINT  (SUBSONIC)
      IPC(I) = IHE
      GO TO 50
   40 CONTINUE
C                         SHOCK POINT
      IPC(I) = IHS
   50 CONTINUE
      WRITE (6,110) J, (IPC(I),I=IUP,IDOWN)
   60 CONTINUE
      RETURN

  100 FORMAT(1H1,9X,18HFLOW CHARACTER MAP/5X,1H(,1H [illegible]//20X,
     1 1H+ = PARABOLIC/20X,1H- = HYPERBOLIC/20X,1HS = SHOCK/20X,
     2 1H- = ELLIPTIC//)
  110 FORMAT(1X,I3,5X,100A1)

      END
```

```
      SUBROUTINE PRTSH(Z,ARG,L,NSHOCK,CDSH,LPRT)
C                 PRINTOUT WAVE DRAG CONTRIBUTION AND TOTAL PRESSURE
C                 LOSS ALONG SHOCK WAVE
C                 CALLED BY - CDCOLL.

      COMMON / COM2/ AK       , ALPHA    , DUB      , GAM1     , RTK
      COMMON /COM18/ CDFACT   , CLFACT   , CMFACT   , CPFACT   , CPSTAR
      LOGICAL          PHYS
      INTEGER          PRTFLG   , SIMDEF
      COMMON /COM27/ CL       , DELTA    , DELRT2   , EMACH    , EMROOT ,
     1                PHYS     , PRTFLG   , SIMDEF   , SONVEL   , VFACT  ,
     2                YFACT
      DIMENSION Z(1), ARG(1)
      CDYCOF = -CDFACT*GAM1/(6.*YFACT)
      POYCOF = DELTA*DELTA*DL*EMACH*GAM1/(GAM1-1.)/12.
      IF(NSHOCK .EQ.1) WRITE(6,1001)
      WRITE(6,1002) NSHOCK, CDSH
      DO 10 K = 1,L
      YY = Z(K)*YFACT
      CDY = CDYCOF*ARG(K)
      POY = 1. + POYCOF*ARG(K)
      WRITE(6,1003) YY,CDY,POY
10    CONTINUE
      IF(LPRT .EQ. 1) WRITE(6,1004)
      RETURN
 1001 FORMAT(1H1,20X,40HSHOCK WAVE DRAG AND TOTAL PRESSURE PROFILE ,
     1 8HOUTPUT/20X,48(1H-)//10X 24HVISCID WAVE PROFILES FOR INDIVIDUAL ,
     2 30HSHOCK WAVES WITHIN MOMENTUM CONTOUR)
 1002 FORMAT(0H0SHOCK,I3//26H WAVE DRAG FOR THIS SHOCK=,F12.6/
     *    6X,1HY,9X,9HCD(Y),8X,6HPO/PO(INF))
 1003 FORMAT(1X,3F12.6)
 1004 FORMAT(38H0SHOCK WAVE EXTENDS OUTSIDE CONTOUR/
     * 43H PRINTOUT OF SHOCK LOSSES ARE NOT AVAILABLE FOR REST OF SHOCK)
      END
```

```
      SUBROUTINE PRTWAL

C                       PRINTS PRESSURE COEFFICIENT AND FLOW ANGLE
C                       ON Y=-H AND Y=+H, AND PLOTS CP ALONG SIDE OF
C                       TABULATION.
C                       CALLED BY - PRINT.

      COMMON           P(100,101),X(100),Y(100)
      COMMON / COM1/ IMIN     , IMAX     , IUP      , IDOWN    , ILE    ,
     1                ITE      , JMIN     , JMAX     , JUP      , JLOW   ,
     2                JTOP     , JBOT
      COMMON / COM3/ XDIFF(100),YDIFF(100)
      COMMON /COM12/ F        , H        , HALFPI   , PI       , RTANPH ,
     1                TWOPI
      COMMON /COM18/ CDFACT   , CLFACT   , CMFACT   , CPFACT   , CPSTAR
      LOGICAL          PHYS
      INTEGER          PRTFLG   , SIMDEF
      COMMON /COM27/ CL       , DELTA    , DELRT2   , EMACH    , EMROOT ,
     1                PHYS     , PRTFLG   , SIMDEF   , SONVEL   , VFACT  ,
     2                YFACT
      INTEGER          BCTYPE
      COMMON /COM28/ BCTYPE   , CIRCFF   , FHINV    , POR      , CIRCTE
      COMMON /COM30/ CPL(100), CPU(100), VL(100), VU(100), RESTI(6)

      DIMENSION  BCT(15) , LINE(60)

      DATA   IB /1H /, IL /1HL/, IU /1HU/, I1 /1H1/, IBB /1HB/
      DATA  BCT /4H    ,4HFREE,4H AIR,4H  SO,4HLID ,4HWALL,4H     ,
     1     4HFREE,4H JET,4HSLOT,4HTED ,4HWALL,4H POR,4HOUS ,4HWALL/
C                                 PRINT SINGLE VARIABLES

      I2 = 3 * BCTYPE
      I1 = I2 - 2
      WRITE (6,900) (BCT(I),I=I1,I2)
      TWH = H * YFACT
      WRITE(6,901) TWH,H
      IF( BCTYPE .LT. 5 )  GO TO 4
      PORF = POR / YFACT
      WRITE(6,903) PORF,POR
    4 CONTINUE
      IF(  BCTYPE .NE. 4  .AND.  BCTYPE .NE. 6 )  GO TO 6
      WRITE (6,904) F
    6 CONTINUE
      WRITE (6,907) CPSTAR

      CPMIN =  1.0E37
      CPMAX = -CPMIN

      CPT = -2.0 * CPFACT
      DO 10 I=IUP,IDOWN
      CPL(I) = CPT   * PX(I,JMIN)
      CPU(I) = CPT   * PX(I,JMAX)
      CPMAX   = AMAX1(CPMAX,CPU(I), CPL(I))
      CPMIN   = AMIN1(CPMIN,CPU(I), CPL(I))
   10 CONTINUE
      DO 20 I=IUP,IDOWN
      IF (BCTYPE .NE. 2) GO TO 11
C                                 SOLID WALL
      VL(I) = 0.0
      VU(I) = 0.0
      GO TO 20
   11 CONTINUE
      IF (BCTYPE .NE. 3) GO TO 12
C                                 FREE JET
      VL(I) = VFACT * PY(I,JMIN)
      VU(I) = VFACT * PY(I,JMAX)
      GO TO 20
   12 CONTINUE
      IF (BCTYPE .NE. 4) GO TO 13
C                                 SLOTTED WALL
      VL(I) =  VFACT * FHINV * (P(JBOT,I) + .75 * CIRCFF)
      VU(I) = -VFACT * FHINV * (P(JTOP,I) - .25 * CIRCFF)
      GO TO 20
   13 CONTINUE
C                                 POROUS WALL
      IF (POR .GT. 1.5) GO TO 14
      VL(I) =  VFACT * POR * XDIFF(I)*(P(JMIN,I)-P(JMIN,I-1))
      VU(I) = -VFACT * POR * XDIFF(I)*(P(JMAX,I)-P(JMAX,I-1))
      GO TO 20
   14 CONTINUE
      VL(I) = VFACT *.25*(P(JMIN+1,I+1)+2.*P(JMIN+1,I)+P(JMIN+1,I-1)
     1                 - P(JMIN  ,I+1)-2.*P(JMIN  ,I)-P(JMIN  ,I-1))
     2                 / (Y(JMIN+1)-Y(JMIN))
      VU(I) = VFACT *.25*(P(JMAX,I+1)  +2.*P(JMAX,I)  +P(JMAX,I-1)
```

Microfilmed ... [illegible]
Best Available Copy

TABLE OF CONTENTS

Section	Page No.	
1. SUMMARY	1	1/A7
2. INTRODUCTION	2	1/A8
3. LIST OF SYMBOLS	5	1/A11
4. GENERAL USER'S MANUAL	8	1/A14
4.1 Description of Program	8	1/A14
4.1.1 Differential equations and transonic scaling	8	1/A14
4.1.2 Geometry	9	1/B1
4.1.3 Boundary conditions	9	1/B1
4.1.4 Aerodynamic variables	13	1/B5
4.1.5 Numerical formulation	13	1/B5
4.1.6 Operating procedure	17	1/B9
4.1.7 Overall program flow chart	19	1/B11
4.2 Program Input	20	1/B12
4.2.1 Dictionary of input variables	20	1/B12
4.2.2 Control options	22	1/B14
4.2.3 Default values	30	1/C8
4.2.4 Input data format	31	1/C9
4.3 Error and Information Messages	34	1/C12
4.4 Program Output	38	1/D2
4.4.1 General description of the output	38	1/D2
4.4.2 Sample cases	41	1/D5
4.5 Comparisons with Data and Other Methods	42	1/D6
4.6 Program Run Times	44	1/D8
4.7 Machine Compatability	46	1/D10
5. PROGRAMMER'S MANUAL	102	2/A11
5.1 Dictionary of Subroutine Variables in Common	102	2/A11
5.2 Description of Individual Subroutines	116	2/B11
5.2.1 Subroutine ANGLE	116	2/B11
⋮ ⋮		
5.2.53 Subroutine VROOTS	145	2/E1
APPENDIX - LISTING OF COMPUTER PROGRAM TSFOIL	146	2/E2
REFERENCES	177	3/B11

```
     1                      - P(JMAX-1,I+1)-2.*P(JMAX-1,I)-P(JMAX-1,I-1))
     2                      / (Y(JMAX)-Y(JMAX-1))
  20 CONTINUE

      CPLARG  = AMAX1(CPMAX,ABS(CPMIN))
      UNPCOL  = CPLARG / 25.

C                        LOCATE CP* FOR PRINTER PLOT

      COL           = -CPSTAR / UNPCOL
      NCOL          = SIGN(ABS(COL)+.5), COL)
      NCOLS         = NCOL + 30

      WRITE (6,210)

C                        PRINT COLUMN HEADERS.

      WRITE (6,220)

      DO 40 I=IUP,IDOWN

      DO 30 K=1,60
      LINE1(K) = IB
   30 CONTINUE

      COL           = -CPUW(I) / UNPCOL
      NCOL          =  SIGN(ABS(COL) + .5) , COL)
      NCOLU         =  NCOL + 30
      LINE1(NCOLU) =  IU

      COL           = -CPLW(I) / UNPCOL
      NCOL          =  SIGN(ABS(COL) + .5) , COL)
      NCOLL         =  NCOL + 30
      LINE1(NCOLL) =  IL
      IF ( NCOLL .EQ. NCOLU ) LINE1(NCOLL) = IBB
      IF (IABS(NCOLS) .LT. 61) LINE1(NCOLS) = IS

      WRITE (6,230) I, X(I), CPLW(I), VLW(I), CPUW(I), VUW(I), LINE1
   40 CONTINUE
      RETURN

  210 FORMAT(1H0,27X,5HLOWER,23X,5HUPPER/20X,4HY=-H,24X,4HY=+H)
  220 FORMAT(3X,1HI,8X,1HX,10X,2HCP,9X,5HTHETA,12X,2HCP,9X,5HTHETA/)
  230 FORMAT(1H ,I3,3F12.6,4X,2F12.6,2X,60A1)
  903 FORMAT(2H1 ,34X,20H BOUNDARY CONDITION.)
  904 FORMAT(1H0,10X,24HH (TUNNEL HALF HEIGHT) =,F9.6,15X,
     1 10HSCALED H =,F9.6)
  905 FORMAT(1H0,11X,23HPOR (POROSITY FACTOR) =,F9.6,15X,
     1 12HSCALED POR =,F9.6)
  906 FORMAT(1H0,14X,20HF (SLOT PARAMETER) =,F9.6)
  907 FORMAT(1H0,29X,5HCP* =,F9.6)

      END


      FUNCTION PX(I,J)
C                   FUNCTION PX COMPUTES U = DP/DX AT POINT I,J
C                   CALLED BY - CDCOLE, DRAG, FINDSK, MACHMP, M1LINE,
C                               NEWISK,  PRINT1, PRTFLD, PRTWAL.

      COMMON            P(102,101),X(100)  , Y(100)
      COMMON / COM1/ IMIN       , IMAX    , IUP     , IDOWN   , ILE     ,
     1               ITE        , JMIN    , JMAX    , JUP     , JLOW    ,
     2               JTOP       , JBOT
      COMMON / COM5/ XDIFF(100),YDIFF(100)

C                   TEST TO LOCATE END POINTS
      IF(I .EQ. IMIN) GO TO 10
      IF(I .EQ. IMAX) GO TO 20
C                   INTERIOR MESH POINT
      PJI = P(J,I)
      PX = .5*(XDIFF(I+1)*(P(J,I+1)-PJI)  + XDIFF(I)*(PJI-P(J,I-1)))
      RETURN
 10   CONTINUE
C                   UPSTREAM BOUNDARY
      PX = 1.5*XDIFF(I+1)*(P(J,I+1)-P(J,I)) -
     1     0.5*XDIFF(I+2)*(P(J,I+2) - P(J,I+1))
      RETURN
 20   CONTINUE
C                   DOWNSTREAM BOUNDARY
      PX = 1.5*XDIFF(I)*(P(J,I) - P(J,I-1))
     1   -0.5*XDIFF(I-1)*(P(J,I-1) - P(J,I-2))
      RETURN
      END
```

```
      FUNCTION PY(I,J)
C                   FUNCTION PY COMPUTES V = DP/DY AT POINT I,J
C                   CALLED BY - CDCOLE, PRTFLD, PRTWAL.

      COMMON            P(102,101),X(100)  , Y(100)
      COMMON / COM1/ IMIN       , IMAX    , IUP     , IDOWN   , ILE     ,
     1               ITE        , JMIN    , JMAX    , JUP     , JLOW    ,
     2               JTOP       , JBOT
      COMMON / COM2/ AK         , ALPHA   , DUB     , GAM1    , RTK
      COMMON / COM5/ XDIFF(100),YDIFF(100)
      COMMON / COM6/ FL(100)    , FXL(100), FU(100) , FXU(100),
     1              CAMBER(100), THICK(100),VOL     , XFOIL(100), IFOIL
      COMMON /COM26/ PJUMP(100)
      INTEGER        BCTYPE
      COMMON /COM28/ BCTYPE     , CIRCFF  , FHINV   , POR     , CIRCTE

C                   TEST FOR END POINTS OR POINTS NEAR AIRFOIL SLIT
      IF(J .EQ. JMIN) GO TO 10
      IF(J .EQ. JLOW) GO TO 20
      IF(J .EQ. JUP ) GO TO 40
      IF(J .EQ. JMAX) GO TO 50
C                   I,J IS AN INTERIOR POINT
      PJI = P(J,I)
      PY = .5*(YDIFF(J+1)*(P(J+1,I)-PJI) + YDIFF(J)*(PJI-P(J-1,I)))
      RETURN
 10   CONTINUE
C                   I,J IS ON LOWER BOUNDARY. USE ONE SIDED DERIVATIVE
      PY = 1.5* YDIFF(J+1)*(P(J+1,I) - P(J,I)) -
     1     0.5* YDIFF(J+2)*(P(J+2,I) - P(J+1,I))
      RETURN
 20   CONTINUE
C                   I,J IS ON ROW OF MESH PIINTS BELOW AIR*OIL.

      VMINUS = YDIFF(J)*(P(J,I) - P(J-1,I))

C                   TEST TO SEE IF I,J IS AHEAD, UNDER, OR BEHIND SLIT.
      IF(I .LT. ILE) GO TO 21
      IF(I .GT. ITE) GO TO 22
C                   I,J IS UNDER AIRFOIL.  USE DERIVATIVE BOUNDARY
C                   CONDITION.
      IC = I - ILE + 1
      PY = .5 * (FXL(IC) - ALPHA + VMINUS)
      RETURN
 21   CONTINUE
C                             I,J IS AHEAD OF AIRFOIL.
      PY = .5*((P(JUP,I) - P(JLOW,I)) * YDIFF(JUP) + VMINUS)
      RETURN
 22   CONTINUE
C                      I,J IS BEHIND AIRFOIL
      PY = .5*((P(JUP,I) - PJUMP(I) - P(JLOW,I)) * YDIFF(JUP) + VMINUS)
      RETURN
 40   CONTINUE
C                   I,J IS ON ROW OF MESH POINTS ABOVE AIRFOIL
      VPLUS = YDIFF(J+1)*(P(J+1,I) - P(J,I))
C                   TEST TO SEE IF I IS AHEAD OF, OVER, OR BEHIND
C                   AIRFOIL SLIT.
      IF(I .LT. ILE) GO TO 41
      IF(I .GT. ITE) GO TO 42
      IC = I - ILE + 1
      PY = .5 * (VPLUS + FXU(IC) - ALPHA)
      RETURN
 41   CONTINUE
C                             I,J IS AHEAD OF AIRFOIL.
      PY = .5*((P(JUP,I) - P(JLOW,I)) * YDIFF(JUP) + VPLUS)
      RETURN
 42   CONTINUE
C                   I,J IS BEHIND AIRFOIL
      PY = .5*((P(JUP,I) - PJUMP(I) - P(JLOW,I)) * YDIFF(JUP) + VPLUS)
      RETURN
 50   CONTINUE
C                   I,J IS ON TOP ROW OF MESH POINTS. USE ONE SIDED
C                   FORMULA.
      PY = 1.5*YDIFF(J)*(P(J,I) - P(J-1,I))
     1  -  0.5*YDIFF(J-1)*(P(J-1,I) - P(J-2,I))

      RETURN
      END
      SUBROUTINE READIN                                                 READIN   2
                                                                        READIN   3
C                 CALLED BY - TSFOIL.                                   READIN   4
                                                                        READIN   5
C                                INPUT EXPLANATION                      READIN   6
C***********************************************************************READIN   7
C                                                                       READIN   8
C        ALL INPUT IS READ IN THIS SUBROUTINE. THE ORDER IS AS DESCRIBED READIN   9
C        BELOW.                                                         READIN  10
C        1.) ONE CARD OF TITLE INFORMATION. AN +A4+ (ALPHANUMERIC) FORMAT READIN  11
C            IS USED TO READ AND WRITE THIS INFORMATION. MULTIPLE CASES READIN  12
C            MAY BE RUN WITH THIS PROGRAM AND THE DATA FOR EACH CASE   READIN  13
C            MUST START WITH THIS CARD. THE LAST CARD OF THE INPUT MUST READIN  14
C            BE A CARD WITH THE WORD +FINISHED+ IN THE FIRST 8 COLUMNS. READIN  15
C                                                                       READIN  16
C        2.) NAMELIST CONTAINING THESE PARAMETERS IS NOW READ. (SEE    READIN  17
C            FORTRAN MANUAL FOR DESCRIPTION OF NAMELIST INPUT). THE    READIN  18
C            BLOCK DATA SUBROUTINE SETS A DEFAULT VALUE, AS NOTED BELOW, READIN  19
C            FOR ALL OF THESE PARAMETERS.                               READIN  20
C            ONLY THE VALUES WHICH ARE DIFFERENT FROM THE PREVIOUS CASE READIN  21
C            (OR DEFAULT) MUST BE INCLUDED, ALTHOUGH AT LEAST ONE VALUE READIN  22
C            MUST BE INPUT BY NAMELIST FOR EACH CASE.                  READIN  23
C                         * (F) = FLOATING POINT *                     READIN  24
C                         * (I) = INTEGER        *                     READIN  25
C                         * (L) = LOGICAL        *                     READIN  26
C                                                         DEFAULT      READIN  27
C                                                          VALUE       READIN  28
C              AMESH  (L) OPTION FOR ANALYTICAL MESH CALC.     .F.      READIN  29
C                         .TRUE. X AND Y MESH VALUES ARE COMPUTED       READIN  30
C                            ACCORDING TO AN ANALYTIC                   READIN  31
C                            FORMULATION. THE GRID SIZE IS              READIN  32
C                            IMAXI=81,JMAXI=64, AND ALLOWS              READIN  33
C                            FOR 1-3 MESH CUTS.                         READIN  34
C                         .FALSE. X AND Y POINTS ARE THE DEFAULT        READIN  35
C                                VALUES OR THE VALUES SUPPLIED          READIN  36
C                                BY THE USER THRU NAMELIST.             READIN  37
C              EMACH  (F) FREESTREAM MACH NUMBER.               .75     READIN  38
C                         NOTE****EMACH MAY NOT BE = 1.0                READIN  39
C              DELTA  (F) BODY THICKNESS RATIO.                 .115    READIN  40
C              ALPHA  (F) ANGLE OF ATTACK (DEGREES IF PHYS=T)   .12     READIN  41
C              AK     (F) TRANSONIC SIMILARITY PARAMETER.       0.0     READIN  42
C                         (INPUT REQUIRED ONLY IF PHYS = .F. )          READIN  43
C              GAM    (F) RATIO OF SPECIFIC HEATS.              1.4     READIN  44
C              SIMDEF (I) SIMILARITY DEFINITION.                  3     READIN  45
C                         =1 COLE                                       READIN  46
C                         =2 SPREITER                                   READIN  47
C                         =3 KRUPP                                      READIN  48
C                         =4 USER                                       READIN  49
C              PRTFLO (I)  OPTION FOR PRINT OF FINAL FLOW FIELD.  1     READIN  50
C                         =1  NO FLOW FIELD PRINT.                      READIN  51
C                         =2  ALL J LINES PRINTED.                      READIN  52
C                         =3 PRINT 3 J LINES AROUND MAXIMUM ERROR       READIN  53
C              PSTART (I) OPTION FOR INITIALIZING P ARRAY.        1     READIN  54
C                         =1 SET TO ZERO.                               READIN  55
C                         =2 READ P FROM UNIT 7                         READIN  56
C                         =3 USE P IN CORE (PREVIOUS CASE).             READIN  57
C              PSAVE  (L) OPTION FOR SAVING RESTART BLOCK OF    .F.     READIN  58
C                           VALUES ON UNIT 3.                           READIN  59
C                         =.TRUE. SAVE FOR RESTART.                     READIN  60
C                         =.FALSE. DO NOT SAVE.                         READIN  61
C              FCR    (L) FULLY CONSERVATIVE RELAXATION OPTION  .T.     READIN  62
C                         =.TRUE. DIFFERENCE EQUATIONS ARE              READIN  63
C                                 FULLY CONSERVATIVE FORM.              READIN  64
C                         =.FALSE. DIFFERENCE EQUATIONS NOT             READIN  65
C                                 CONSERVATIVE AT SHOCK WAVES.          READIN  66
C              KUTTA  (L) KUTTA CONDITION OPTION.               .T.     READIN  67
C                         =.TRUE. KUTTA CONDITION IS ENFORCED           READIN  68
C                         =.FALSE. LIFT COEFFICIENT SPECIFIED           READIN  69
C                                 BY USER.                              READIN  70
C              CLSET  (F) LIFT COEFFICIENT, USED IF KUTTA       .0      READIN  71
C                         IS FALSE.                                     READIN  72
C              BCFOIL (I) OPTION FOR FOIL OR BODY GEOMETRY.       3     READIN  73
C                         =1 NACA00XX                                   READIN  74
C                         =2 PARABOLIC ARC.                             READIN  75
C                         =3 ORDINATES (READ LATER IN NAMELIST          READIN  76
C                            IF DIFFERENT FROM DEFAULT VALUES WHICH     READIN  77
C                            ARE FOR THE KORN AIRFOIL).                 READIN  78
C                         =4 USER OPTION.                               READIN  79
C              BCTYPE (I) DESCRIBES THE TYPE OF FLOW TO BE        1     READIN  80
C                         COMPUTED.                                     READIN  81
C                         =1 FREE AIR.                                  READIN  82
C                         =2 SOLID WALL.                                READIN  83
C                         =3 FREE JET.                                  READIN  84
C                         =4 SLOTTED WALL.                              READIN  85
C                         =5 POROUS WALL.                               READIN  86
C              F      (F) TUNNEL SLOT PARAMETER.                0.      READIN  87
C              H      (F) TUNNEL HALF HEIGHT/CHORD RATIO.       0.      READIN  88
C              POR    (F) WALL POROSITY FACTOR.                 .0      READIN  89
C              PHYS   (L) TYPE OF SCALING TO USE FOR I/O.       .T.     READIN  90
C                         =.TRUE. PHYSICALLY SCALED VALUES.             READIN  91
C                         =.FALSE. TRANSONICALLY SCALED VALUES          READIN  92
C                         FOR PHYS = .F. , ALSO INPUT VALUE FOR AK.     READIN  93
C              IMAXI  (I) NUMBER OF X-MESH POINTS(.LE. 100)     77      READIN  94
C              JMAXI  (I) NUMBER OF Y-MESH POINTS(.LE. 100)     56      READIN  95
C              IMIN   (I) X MESH POINT WHERE CALC IS TO START     1     READIN  96
C              JMIN   (I) Y MESH POINT WHERE CALC IS TO START     1     READIN  97
C              ICUT   (I) CONTROL FOR MESH CUT AND REFINEMENT.    2     READIN  98
C                         = 0  INPUT MESH IS USED TO CONVERGENCE.       READIN  99
C                         = 1  INPUT MESH MAY BE CUT ONCE.              READIN 100
C                         = 2  INPUT MESH MAY BE CUT TWICE.             READIN 101
C              WE      (F)  3 VALUES FOR RELAXATION FACTOR FOR 1.8      READIN 102
C                           ELIPTIC PTS. 1-ST FOR COARSE MESH, 1.9      READIN 103
C                           2-ND FOR MED. MESH AND 3-RD FOR    1.95     READIN 104
C                           FINE MESH. DEFAULT VALUES ARE SUGGESTED     READIN 105
C                          VALUES, IN ORDER. IF SPECIFIED IN INPUT      READIN 106
C                          ALL THREE VALUES MUST BE GIVEN.              READIN 107
C              WCIRC  (F) RELAXATION FACTOR FOR CIRCULATION.    1.0     READIN 108
C              MAXIT   (I) MAXIMUM NUMBER OF ITERATION CYCLES  500      READIN 109
C                         ALLOWED.                                      READIN 110
C              CVERGE (F) CONVERGENCE CRITERION FOR RESIDUALS .00001    READIN 111
C                         OF P.                                         READIN 112
C              DVERGE (F) DIVERGENCE CRITERION FOR RESIDUALS    10.     READIN 113
C                         OF P.                                         READIN 114
C              RIGF   (F) REIGLES RULE FOR BODY SLOPE.          0.0     READIN 115
C              EPS     (F) COEFFICIENT OF PXT                    .2     READIN 116
C              IPRTER (I) CONTROL FOR FREQUENCY OF PRINT OF      10     READIN 117
C                         LINE IN MESH WHERE ERROR IS LARGEST.          READIN 118
C                         I.E. IPRTER=10 , LINE WILL BE PRINTED         READIN 119
C                              EVERY 10-TH ITERATION.                   READIN 120
C                                                                       READIN 121
C NOTE***** WHEN ARRAYS ARE READ BY NAMELIST THE FULL ARRAY MUST BE     READIN 122
C           SET, I.E. IF ALL VALUES ARE NOT REQUIRED THE ARRAY MAY      READIN 123
C           BE FILLED USING MULTIPLE ZEROS. (N*0.0)                     READIN 124
C              XU     (F) ARRAY - X VALUES FOR UPPER BODY. USED         READIN 125
C                             IF BCFOIL = 3. KORN AIRFOIL USING ALL     READIN 126
C                             100 PTS.(UPPER) AND 75 (LOWER) IS DEFAULT. READIN 127
C              XL     (F) ARRAY - X VALUES FOR LOWER BODY.              READIN 128
C              YU     (F) ARRAY - Y VALUES FOR UPPER BODY.              READIN 129
C              YL     (F) ARRAY - Y VALUES FOR LOWER BODY.              READIN 130
C              NU     (I) NUMBER OF POINTS TO USE FOR UPPER BODY 100    READIN 131
C              NL     (I) NUMBER OF POINTS TO USE FOR LOWER BODY  75    READIN 132
                                                                        READIN 133
C****** NOTE THIS PROGRAM USES A MESH REFINEMENT METHOD FOR DECREASING   READIN 134
C           COMPUTER TIME. FOLLOW THE RULES BELOW FOR CONSTRUCTING THE  READIN 135
C           X AND Y MESH TO TAKE FULL ADVANTAGE OF THIS FEATURE.        READIN 136
C               IMAXI - ITE    SHOULD BE A MULTIPLE OF 4.               READIN 137
C               ITE   - IMIN   SHOULD BE A MULTIPLE OF 4.               READIN 138
C               JMAXI - JUP  + 1 SHOULD BE A MULTIPLE OF 4              READIN 139
C               JLOW  - JMIN + 1 SHOULD BE A MULTIPLE OF 4              READIN 140
C                          (WHERE JLOW IS LAST POINT BELOW SLIT AND     READIN 141
C                                 JUP IS FIRST POINT ABOVE SLIT.)       READIN 142
C             WHERE ITE = I FOR X = 1.0 (OR POINT ON BODY CLOSEST TO    READIN 143
C                                       X = 1.0).                       READIN 144
C           SUBROUTINE CKMESH INSPECTS THE X AND Y MESHES TO SEE IF     READIN 145
C           THIS IS TRUE AND, IF NOT, WILL MODIFY INPUT MESH IN SOME    READIN 146
C           CASES.                                                      READIN 147
                                                                        READIN 148
C              XIN    (F) ARRAY - X MESH POINTS. LIMIT 100 PTS.         READIN 149
C              YIN    (F) ARRAY - Y MESH POINTS. LIMIT 100 PTS.         READIN 150
C                         X AND Y MESH DEFAULT VALUES ARE KRUPP         READIN 151
C                                  BASIC GRID.                          READIN 152
                                                                        READIN 153
      COMMON          P(102,101),X(100)  , Y(100)                       BLANK    2
      COMMON / COM1/ IMIN     , IMAX    , IUP     , IDOWN   , ILE     , COM1     2
     1               ITE      , JMIN    , JMAX    , JUP     , JLOW    , COM1     3
     2               JTOP     , JBOT                                    COM1     4
      COMMON / COM2/ AK       , ALPHA   , DUB     , GAM1    , RTK       COM2     2
      LOGICAL        ABORT                                              COM3     2
      COMMON / COM3/ IREF     , ABORT   , ICUT    , KSTEP               COM3     3
      LOGICAL        AMESH                                              COM4     2
      COMMON / COM4/ XIN(100) , YIN(100), AMESH                         COM4     3
      COMMON / COM8/ CVERGE   , DVERGE  , IPRTER  , MAXIT   ,           COM8     2
     1               WE(3)    , EPS                                     COM8     3
      INTEGER        BCFOIL                                             COM9     2
      COMMON / COM9/ BCFOIL   , NL      , NU      , XL(100) , XU(100) , COM9     3
     1               YL(100)  , YU(100) , RIGF                          COM9     4
      LOGICAL XGRDIN,YGRDIN                                             COM10    2
      COMMON /COM10/ YFREE(100) , YTUN(100) , XKRUPP(100) , GAM     ,   COM10    3
     1               JMXF     , JMXT    , XGRDIN  , YGRDIN              COM10    4
      INTEGER        PSTART                                             COM11    2
      LOGICAL        PSAVE                                              COM11    3
      COMMON /COM11/ ALPHAO   , CLOLD   , DELTAO  , DUBO    , EMACHO  ,  COM11    4
     1               IMINO    , IMAXO   , IMAXI   , JMINO   , JMAXO   ,  COM11    5
     2               JMAXI    , PSAVE   , PSTART  , TITLE(8), TITLEO(8), COM11    6
     3               VOLO     , XOLD(100),YOLD(100)                      COM11    7
      COMMON /COM12/ F        , H       , HALFPI  , PI      , RTKPOR  ,  COM12    2
     1               TWOPI                                               COM12    3
      LOGICAL        FCR      , KUTTA                                    COM14    2
      COMMON /COM14/ CLSET    , FCR     , KUTTA   , WCIRC                COM14    3
      LOGICAL        PHYS                                                COM27    2
      INTEGER        PRTFLD   , SIMDEF                                   COM27    3
      COMMON /COM27/ CL       , DELTA   , DELRT2  , EMACH   , EMROOT  ,  COM27    4
     1               PHYS     , PRTFLD  , SIMDEF  , SONVEL  , VFACT   ,  COM27    5
     2               YFACT                                               COM27    6
      INTEGER        BCTYPE                                              COM28    2
      COMMON /COM28/ BCTYPE   , CIRCFF  , FHINV   , POR     , CIRCTE     COM28    3
                                                                         READIN 167
      DATA  DONE /8HFINISHED  /                                          READIN 168
      DATA   IFIRST /0/                                                  READIN 169
      NAMELIST /INP/ AK     , ALPHA  , AMESH  , BCFOIL , BCTYPE , CLSET, READIN 170
     1               CVERGE , DELTA  , DVERGE , EMACH  , EPS    , F    , READIN 171
     2               FCR    , GAM    , H      , ICUT   , IMAXI  , IMIN , READIN 172
     3               IPRTER , JMAXI  , JMIN   , KUTTA  , MAXIT  , NL   , READIN 173
     4               NU     , PHYS   , POR    , PRTFLD , PSAVE  ,        READIN 174
     5               PSTART , RIGF   , SIMDEF , WCIRC  , WE     ,        READIN 175
     6               XIN    , YIN    , XL     , YL     , XU     , YU   , READIN 176
     7               XGRDIN , YGRDIN                                     READIN 177
    5 CONTINUE                                                           READIN 178
      IF (IFIRST .NE. 0) GO TO 2                                         READIN 179
      IFIRST = 1                                                         READIN 180
      CALL SECOND (TIME2)                                                READIN 181
    2 CONTINUE                                                           READIN 182
      TIME1 = TIME2                                                      READIN 183
      CALL SECOND (TIME2)                                                READIN 184
      ELPTM = TIME2 - TIME1                                              READIN 185
      IF (ELPTM .LT. .01) GO TO 3                                        READIN 186
      WRITE (6,932) ELPTM                                                READIN 187
    3 CONTINUE                                                           READIN 188
      READ (5,900) TITLE                                                 READIN 189
      WRITE(6,901) TITLE                                                 READIN 190
      IF (TITLE(1) .NE. DONE) GO TO 10                                   READIN 191
      WRITE(6,904)                                                       READIN 192
      STOP                                                               READIN 193
   10 CONTINUE                                                           READIN 194
      READ (5,INP)                                                       READIN 195
      IF (PSTART .NE. 3) GO TO 13                                        READIN 196
C                             TEST TO SEE IF P ARRAY IN CORE IS USEABLE. READIN 197
      IF ( .NOT. ABORT) GO TO 13                                         READIN 198
      WRITE (6,903)                                                      READIN 199
      GO TO 5                                                            READIN 200
   13 CONTINUE                                                           READIN 201
      IF ( PHYS ) AK = 0.0                                               READIN 202
      IF ( .NOT. AMESH ) GO TO 14                                        READIN 203
      CALL AYMESH                                                        READIN 204
      GO TO 22                                                           READIN 205
   14 CONTINUE                                                           READIN 206
      IF (YGRDIN) GO TO 18                                               READIN 207
C                             IF YIN NOT READ IN NAMELIST, FILL YIN BY   READIN 208
C                             DEFAULT VALUE FOR TUNNEL OR FREE AIR CASE. READIN 209
      IF (BCTYPE .NE. 1) GO TO 16                                        READIN 210
      JMAXI = JMXF                                                       READIN 211
      DO 15 J=JMIN,JMAXI                                                 READIN 212
      YIN(J) = YFREE(J)                                                  READIN 213
   15 CONTINUE                                                           READIN 214
      GO TO 18                                                           READIN 215
   16 CONTINUE                                                           READIN 216
      JMAXI = JMXT                                                       READIN 217
      DO 17 J=JMIN,JMAXI                                                 READIN 218
      YIN(J) = YTUN(J)                                                   READIN 219
   17 CONTINUE                                                           READIN 220
   18 CONTINUE                                                           READIN 221
      IF (XGRDIN) GO TO 22                                               READIN 222
      DO 21 I=IMIN,IMAXI                                                 READIN 223
   21 XIN(I)=XKRUPP(I)                                                   READIN 224
   22 CONTINUE                                                           READIN 225
      WRITE (6,921)                                                      READIN 226
C                                                                        READIN 227
      WRITE (6,906) EMACH, POR, IMIN, BCTYPE, AMESH                      READIN 228
      WRITE (6,907) DELTA, CLSET, IMAXI, BCFOIL, PHYS                    READIN 229
      WRITE (6,908) ALPHA, EPS  , JMIN , PSTART, PSAVE                   READIN 230
      WRITE (6,909) AK   , RIGF , JMAXI, PRTFLD, KUTTA                   READIN 231
      WRITE (6,910) GAM, WCIRC, MAXIT, IPRTER, FCR                       READIN 232
      WRITE (6,911) F, CVERGE, NU, SIMDEF                                READIN 233
      WRITE (6,912) H, DVERGE, NL, ICUT                                  READIN 234
      WRITE (6,920) WE                                                   READIN 235
      WRITE (6,913)                                                      READIN 236
      WRITE (6,919) (XIN(I),I=IMIN,IMAXI)                                READIN 237
```

```
      WRITE (6,914)                                                      READIN 238
      WRITE (6,919) (YIN(J),J=JMIN,JMAXI)                                READIN 239
      IF (BCFOIL .LE. 2) GO TO 19                                        READIN 240
      WRITE (6,915)                                                      READIN 241
      WRITE (6,919) (XU(I),I=1,NU)                                       READIN 242
      WRITE (6,916)                                                      READIN 243
      WRITE (6,919) (YU(I),I=1,NU)                                       READIN 244
      WRITE (6,917)                                                      READIN 245
      WRITE (6,919) (XL(I),I=1,NL)                                       READIN 246
      WRITE (6,918)                                                      READIN 247
      WRITE (6,919) (YL(I),I=1,NL)                                       READIN 248
   19 CONTINUE                                                           READIN 249
      GAM1 = GAM + 1.0                                                   READIN 250
      IREF = 0                                                           READIN 251
      IMAX = IMAXI                                                       READIN 252
      JMAX = JMAXI                                                       READIN 253
      IM1  = IMAX - 1                                                    READIN 254
      JM1  = JMAX - 1                                                    READIN 255
      IF (IMAXI .GT. 100 .OR.  JMAXI .GT. 100) CALL INPERR (1)           READIN 256
C                             ANY CALL TO INPERR CAUSES A MESSAGE TO BE  READIN 257
C                             PRINTED AND EXECUTION IS STOPPED.          READIN 258
                                                                         READIN 259
C                             CHECK INPUT MESH FOR MONOTONICALLY INCREASING READIN 260
C                             VALUES.                                    READIN 261
      DO 20 I=IMIN,IM1                                                   READIN 262
      IF (XIN(I) .GE. XIN(I+1)) CALL INPERR (2)                          READIN 263
   20 CONTINUE                                                           READIN 264
                                                                         READIN 265
      DO 30 J=JMIN,JM1                                                   READIN 266
      IF (YIN(J) .GE. YIN(J+1)) CALL INPERR (3)                          READIN 267
   30 CONTINUE                                                           READIN 268
                                                                         READIN 269
      IF (EMACH .LT.    .5 .OR. EMACH .GT.  2.0) CALL INPERR (4)         READIN 270
      IF (ALPHA .LT. -9.0  .OR. ALPHA .GT. 9.0 ) CALL INPERR (5)         READIN 271
      IF (DELTA .LT.   0.0 .OR. DELTA .GT.  1.0) CALL INPERR (6)         READIN 272
                                                                         READIN 273
                                                                         READIN 274
C                             COMPUTE ILE AND ITE (LEADING AND TRAILING EDGE) READIN 275
                                                                         READIN 276
      CALL ISLIT ( XIN )                                                 READIN 277
                                                                         READIN 278
C                             COMPUTE JLOW AND JUP (LOCATION OF BODY SLIT) READIN 279
                                                                         READIN 280
      CALL JSLIT ( YIN )                                                 READIN 281
                                                                         READIN 282
C                                                                        READIN 283
C                             CHECK NUMBER OF MESH POINTS, IF NOT ODD ADD READIN 284
C                             POINTS TO APPROPRIATE AREAS TO MAKE ODD NO. READIN 285
C                                                                        READIN 286
      CALL CKMESH                                                        READIN 287
                                                                         READIN 288
C                             CHECK BOUNDS OF YMESH FOR TUNNEL CALCULATIONS. READIN 289
      IF (BCTYPE .EQ. 1) GO TO 90                                        READIN 290
      HTM = H - .00001                                                   READIN 291
      HTP = H + .00001                                                   READIN 292
      YS  = ABS(YIN(JMIN))                                               READIN 293
      YE  = ABS(YIN(JMAX))                                               READIN 294
      IF (YS .LT. HTM .OR. YS .GT. HTP) GO TO 40                         READIN 295
      IF (YE .GE. HTM .AND. YE .LE. HTP) GO TO 90                        READIN 296
   40 CONTINUE                                                           READIN 297
C                             RESCALE Y MESH TO -H,+H BOUNDS.            READIN 298
      TERM = -H / YIN(JMIN)                                              READIN 299
      DO 45 J=JMIN,JLOW                                                  READIN 300
      YIN(J) = TERM * YIN(J)                                             READIN 301
   45 CONTINUE                                                           READIN 302
      TERM = H / YIN(JMAX)                                               READIN 303
      DO 50 J=JUP,JMAX                                                   READIN 304
      YIN(J) = TERM * YIN(J)                                             READIN 305
   50 CONTINUE                                                           READIN 306
                                                                         READIN 307
   90 CONTINUE                                                           READIN 308
C                             IF PSTART = 2 READ OLD VALUES FROM TAPE 7. READIN 309
      IF (PSTART .NE. 2) GO TO 100                                       READIN 310
                                                                         READIN 311
      REWIND 7                                                           READIN 312
      READ (7,900) TITLEO                                                READIN 313
      READ  (7,902) IMAXO, JMAXO, IMINO, JMINO                           READIN 314
      READ  (7,903) CLOLD, EMACHO, ALPHAO, DELTAO, VOLO, DUBO            READIN 315
      READ (7,903) (XOLD(I),I=IMINO,IMAXO)                               READIN 316
      READ (7,903) (YOLD(J),J=JMINO,JMAXO)                               READIN 317
      DO 60 I=IMINO,IMAXO                                                READIN 318
      READ (7,903) (P(J,I), J=JMINO,JMAXO)                               READIN 319
   60 CONTINUE                                                           READIN 320
      WRITE(6,1000) TITLEO, IMINO,IMAXO,JMINO,JMAXO,CLOLD,EMACHO,        READIN 321
     1               ALPHAO,DELTAO,VOLO, DUBO                            READIN 322
```

```
364 CONTINUE
    RETURN

900 FORMAT(8A10)
901 FORMAT(1H1,23X,8A10)
902 FORMAT(4I5)
903 FORMAT(8F10.0)
904 FORMAT(1H1)
905 FORMAT(21H0 CALCULATION ABORTED//
   1  43H OUTPUT OF PREVIOUS SOLUTION NOT AVAILABLE.)
906 FORMAT(1H0,12X,7HEMACH =,F9.5,6X,5HPOR =,F9.5,7X,6HIMIN =,I5,
   1     5X,8HBCTYPE =,I3,9X,8HAMESH = ,L1)
907 FORMAT(1H0,12X,7HDELTA =,F9.5,6X,7HCLSET =,F9.5,6X,7HIMAXI =,I5,
   1     3X,8HPCFOIL =,I3,10X,7HPHYS = ,L1)
908 FORMAT(1H0,12X,7HALPHA =,F9.5,6X,5HEPS =,F9.5,7X,6HJMIN =,I5,
   1     5X,8HPSTART =,I3,9X,8HPSAVE = ,L1)
909 FORMAT(1H0,13X,4HMAX =,F9.5,7X,6HRIGF =,F9.5,6X,7HJMAXI =,I5,
   1     5X,8HPRTFLD =,I3,9X,8HKUTTA = ,L1)
910 FORMAT(1H0,14X,3HGAM =,F9.5,6X,7HWCIRC =,F9.5,6X,7HMAXIT =,I5,
   1     5X,8HIPRTER =,I3,11X,6HFCR = ,L1)
911 FORMAT(1H0,16X,3HF =,F9.5,5X,8HCVERGE =,F9.5,9X,4HNU =,I5,
   1     5X,8HSIMDEF =,I3)
912 FORMAT(1H0,16X,3HH =,F9.5,5X,8HDVERGE =,F9.1,9X,4HNL =,I5,
   1     7X,8HICUT =,I3)
913 FORMAT(1H0,14X,3HXIN)
914 FORMAT(1H0,14X,3HYIN)
915 FORMAT(1H0,15X,2HXU)
916 FORMAT(1H0,15X,2HYU)
917 FORMAT(1H0,15X,2HXL)
918 FORMAT(1H0,15X,2HYL)
919 FORMAT(10X,8F11.6)
920 FORMAT(1H0,15X,5HWE = ,F4.2,2(1H,,F4.2))
921 FORMAT(/////50X,16HINPUT PARAMETERS/50X,16(1H*)/)
930 FORMAT(22H1TIME TO RUN CASE WAS ,F6.2,9H SECONDS.)
1000 FORMAT(39H1P INITIALIZED FROM PREVIOUS RUN TITLED/
   1         1X,8A10/31H WHICH HAD THE FOLLOWING VALUES/
   1         8H IMIN  =,I4/8H IMAX  =,I4/8H JMIN  =,I4/8H JMAX  =,I4/
   1         8H CL    =,F12.8/8H EMACH =,F12.8/8H ALPHA =,F12.8/
   1         8H DELTA =,F12.8/8H VOL   =,F12.8/8H DUB   =,F12.8/)

     END
```

```
      SUBROUTINE RECIRC
C                 SUBROUTINE RECIRC COMPUTES THE FOLLOWING
C                 1.) JUMP IN P AT TRAILING EDGE = CIRCTE
C                 2.) CIRCULATION FOR FARFIELD BOUNDARY = CIRCFF
C                 3.) JUMP IN P ALONG SLIT Y=0, X .GT.1 BY LINEAR
C                     INTERPOLATION BETWEEN CIRCTE AND CIRCFF
C                 4.) P(JO,ITE) AND P(JO, ITE-1)
C                 CALLED BY - SOLVE.

      COMMON          P(102,101),X(100)  , Y(100)
      COMMON / COM1/ IMIN     , IMAX     , IUP      , IDOWN    , ILE      ,
     1               ITE      , JMIN     , JMAX     , JUP      , JLOW     ,
     2               JTOP     , JBOT
      COMMON / COM2/ AK       , ALPHA    , DUB      , GAM1     , RTK
      COMMON / COM7/ CJUP     , CJUP1    , CJLOW    , CJLOW1
      COMMON /COM13/ CDFACT   , CLFACT   , CMFACT   , CPFACT   , CPSTAR
      LOGICAL        FCR      , KUTTA
      COMMON /COM14/ CLSET    , FCR      , KUTTA    , WCIRC
      LOGICAL        OUTERR
      COMMON /COM18/ ERROR    , I1       , I2       , IERROR   , JERROR   ,
     1               OUTERR   , EMU(100,2)          , VC(100)  ,
     2               PI       , DCIRC    , POLD(100,2)
      COMMON /COM22/ CXC(100) , CXL(100), CXR(100), CXXC(100),CXXL(100),
     1               CXXR(100), C1(100)
      COMMON /COM26/ PJUMP(100)
      INTEGER        BCTYPE
      COMMON /COM20/ BCTYPE   , CIRCFF   , FHINV    , POR      , CIRCTE

C                 COMPUTE JUMP IN POTENTIAL AT TRAILING EDGE

      CTEOLD=CIRCTE
      PUP = CJUP*P(JUP,ITE) - CJUP1*P(JUP+1,ITE)
      PLOW = CJLOW*P(JLOW,ITE) - CJLOW1*P(JLOW-1,ITE)
      CIRCTE = PUP - PLOW
C                 COMPUTE FAR FIELD CIRCULATION
      CIRCO = CIRCFF
      IF(KUTTA) CIRCFF = (1.-WCIRC)*CIRCO + CIRCTE*WCIRC
      IF(.NOT. KUTTA) CIRCFF = .5*CLSET/CLFACT
C                 FIX JUMP IN P AT AIRFOIL TRAILING EDGE IF KUTTA=.F.
C                 AND LIFT OF AIRFOIL EXCEEDS CLSET.  THIS TREATMENT
C                 ASSUMES  THAT CL IS INCREASING FROM BELOW CLSET OR
C                 IS NOT TOO MUCH ABOVE CLSET TO START CALCULATION.

      IF ( .NOT. KUTTA) CIRCTE = CIRCFF
      DCIRC=CIRCTE-CTEOLD

C                 SET JUMP IN P ALONG Y = 0, X .GT. 1
      FACTOR = (CIRCFF - CIRCTE)/(X(IMAX) - 1.)
      DO 35 I = ITE,IMAX
      PJUMP(I) = CIRCTE + (X(I) - 1.) * FACTOR
 35   CONTINUE

      RETURN
      END
```

```
      SUBROUTINE REDUB
C                 SUBROUTINE REDUB COMPUTES DOUBLET STRENGTH
C                 FOR LIFTING FREE AIR FLOWS. DOUBLET STRENGTH IS SET
C                 EQUAL TO MODEL VOLUME.  FOR OTHER FLOWS, THE NON
C                 LINEAR CONTRIBUTION IS ADDED.
C                 CALLED BY - SOLVE.

      COMMON          P(102,101),X(100)  , Y(100)
      COMMON / COM1/ IMIN     , IMAX     , IUP      , IDOWN    , ILE      ,
     1               ITE      , JMIN     , JMAX     , JUP      , JLOW     ,
     2               JTOP     , JBOT
      COMMON / COM2/ AK       , ALPHA    , DUB      , GAM1     , RTK
      COMMON / COM3/ XDIFF(100),YDIFF(100)
      COMMON / COM6/ FL(100)  , FXL(100), FU(100) , FXU(100),
     1              CAMBER(100), THICK(100),VOL     , XFOIL(100), IFOIL
      INTEGER         BCTYPE
      COMMON /COM20/ BCTYPE   , CIRCFF   , FHINV    , POR      , CIRCTE
      COMMON /COM30/ XI(100)  , ARG(100)  , REST(204)

      IF(BCTYPE .NE. 1) GO TO 10
      IF(BCTYPE .EQ. 1 .AND. ABS(CIRCFF) .LT. .0001) GO TO 10
      DUB = VOL
      RETURN
 10   CONTINUE
C
C                 COMPUTE  DOUBLE INTEGRAL OF U*U OVER MESH DOMAIN FOR
C                 DOUBLET STRENGTH
C                 U = PX IS CENTERED MIDWAY BETWEEN X MESH POINTS.
C                 FIRST THE INTEGRAL (PX**2)DY IS CALCULATED FOR X
C                 HELD CONSTANT. THUS 1./(X(I+1)-X(I))**2 MAY BE
C                 PULLED OUT OF THE INTEGRAL WHICH IS CALCULATED BY
C                 THE TRAPEZOIDAL RULE. THE X INTEGRATION CORRESPONDS
C                 TO SUMMING THESE INTEGRALS, WHICH LIE MIDWAY BETWEEN
C                 X MESH POINTS, USING A MODIFIED TRAPEZOIDAL RULE.
C
      IEND = IMAX - 1
      DBLSUM = 0.
      DO 50 I=IMIN,IEND
      NARG = 0
      DO 30 J = JMIN, JMAX
      NARG = NARG + 1
      TEMP = P(J,I+1) - P(J,I)
      ARG(NARG) = TEMP * TEMP
      XI(NARG) = Y(J)
 30   CONTINUE
      CALL TRAP(XI,ARG,NARG,SUM)
      DBLSUM = DBLSUM + SUM * XDIFF(I+1)
 50   CONTINUE
      DBLSUM = GAM1*.25*DBLSUM
      DUB = VOL + DBLSUM
      RETURN
      END
      SUBROUTINE REFINE                                                 REFINE   2
C                   ROUTINE TO EXPAND THE X-MESH AND Y-MESH TO          REFINE   3
C                   DOUBLE THE NUMBER OF POINTS IN EACH. WILL HAVE      REFINE   4
C                   TO BE CALLED TWICE IF MESH WAS HALVED TWICE.        REFINE   5
C                   THE P(J,I) MESH IS ALSO FILLED BY                   REFINE   6
C                   INTERPOLATION.                                      REFINE   7
C                   CALLED BY - TSFOIL.                                 REFINE   8
                                                                        REFINE   9
      COMMON        P(102,101),X(100)  , Y(100)                         BLANK    2
      COMMON / COM1/ IMIN     , IMAX    , IUP     , IDOWN   , ILE    ,  COM1     2
     1              ITE      , JMIN    , JMAX    , JUP     , JLOW   ,  COM1     3
     2              JTOP     , JBOT                                     COM1     4
      LOGICAL       ABORT                                               COM3     2
      COMMON / COM3/ IREF     , ABORT   , ICUT    , KSTEP               COM3     3
      LOGICAL       AMESH                                               COM4     2
      COMMON / COM4/ XIN(100) , YIN(100), AMESH                         COM4     3
      COMMON /COM20/ XMID(100), YMID(100)                               COM20    2
      INTEGER       BCTYPE                                              COM28    2
      COMMON /COM28/ BCTYPE   , CIRCFF  , FHINV   , POR     , CIRCTE    COM28    3
      COMMON /COM30/ PT(100)  , REST(304)                               REFINE  16
                                                                        REFINE  17
      JMAXO = JMAX                                                      REFINE  18
      IMAX = 2 * (IMAX - IMIN) + IMIN                                   REFINE  19
      JMAX = 2 * (JMAX - JMIN) + JMIN + 1                               REFINE  20
      IM2  = IMAX - 2                                                   REFINE  21
      JM2  = JMAX - 2                                                   REFINE  22
      IF (IREF .GT. 1) GO TO 30                                         REFINE  23
      DO 10 I=IMIN,IMAX                                                 REFINE  24
      X(I) = XIN(I)                                                     REFINE  25
   10 CONTINUE                                                          REFINE  26
      DO 20 J=JMIN,JMAX                                                 REFINE  27
      Y(J) = YIN(J)                                                     REFINE  28
   20 CONTINUE                                                          REFINE  29
      IREF = 0                                                          REFINE  30
      GO TO 60                                                          REFINE  31
   30 CONTINUE                                                          REFINE  32
      DO 40 I=IMIN,IMAX                                                 REFINE  33
      X(I) = XMID(I)                                                    REFINE  34
   40 CONTINUE                                                          REFINE  35
      DO 50 J=JMIN,JMAX                                                 REFINE  36
      Y(J) = YMID(J)                                                    REFINE  37
   50 CONTINUE                                                          REFINE  38
      IREF = 1                                                          REFINE  39
   60 CONTINUE                                                          REFINE  40
      CALL ISLIT ( X )                                                  REFINE  41
      CALL JSLIT ( Y )                                                  REFINE  42
C                   SPREAD P(J,I) TO ALTERNATE I(X-MESH) POINTS.        REFINE  43
      DO 90 J=JMIN,JMAXO                                                REFINE  44
      K = IMIN - 1                                                      REFINE  45
      DO 70 I=IMIN,IMAX,2                                               REFINE  46
      K = K + 1                                                         REFINE  47
      PT(I) = P(J,K)                                                    REFINE  48
   70 CONTINUE                                                          REFINE  49
      DO 80 I=IMIN,IMAX,2                                               REFINE  50
      P(J,I) = PT(I)                                                    REFINE  51
   80 CONTINUE                                                          REFINE  52
   90 CONTINUE                                                          REFINE  53
C                   SPREAD P(J,I) TO ALTERNATE J (Y-MESH) POINTS.       REFINE  54
      DO 130 I=IMIN,IMAX,2                                              REFINE  55
      K = JMIN - 1                                                      REFINE  56
      JE = JLOW - 1                                                     REFINE  57
      JL = JLOW - 2                                                     REFINE  58
      DO 95 J=JMIN,JE, 2                                                REFINE  59
      K = K + 1                                                         REFINE  60
      PT(J) = P(K,I)                                                    REFINE  61
   95 CONTINUE                                                          REFINE  62
      JST = JUP + 1                                                     REFINE  63
      DO 100 J=JST,JMAX,2                                               REFINE  64
      K = K + 1                                                         REFINE  65
      PT(J) = P(K,I)                                                    REFINE  66
  100 CONTINUE                                                          REFINE  67
      DO 110 J=JMIN,JE,2                                                REFINE  68
      P(J,I) = PT(J)                                                    REFINE  69
  110 CONTINUE                                                          REFINE  70
      DO 120 J=JST,JMAX,2                                               REFINE  71
      P(J,I) = PT(J)                                                    REFINE  72
  120 CONTINUE                                                          REFINE  73
  130 CONTINUE                                                          REFINE  74
C                   INTERPOLATE TO FILL IN THE MISSING P VALUES.        REFINE  75
      DO 140 I=IMIN,IM2                                                 REFINE  76
      PT(I) = (X(I+1)-X(I)) / (X(I+2)-X(I))                             REFINE  77
  140 CONTINUE                                                          REFINE  78
      DO 150 J=JMIN,JE,2                                                REFINE  79
      DO 145 I=IMIN,IM2,2                                               REFINE  80
      P(J,I+1) = P(J,I) + PT(I) * (P(J,I+2) - P(J,I))                   REFINE  81
  145 CONTINUE                                                          REFINE  82
  150 CONTINUE                                                          REFINE  83
      DO 160 J=JST,JMAX,2                                               REFINE  84
      DO 155 I=IMIN,IM2,2                                               REFINE  85
      P(J,I+1) = P(J,I) + PT(I) * (P(J,I+2) - P(J,I))                   REFINE  86
  155 CONTINUE                                                          REFINE  87
  160 CONTINUE                                                          REFINE  88
      DO 170 J=JMIN,JM2                                                 REFINE  89
      PT(J) = (Y(J+1)-Y(J)) / (Y(J+2)-Y(J))                             REFINE  90
  170 CONTINUE                                                          REFINE  91
      DO 190 I=IMIN,IMAX                                                REFINE  92
      DO 175 J=JMIN,JL,2                                                REFINE  93
      P(J+1,I) = P(J,I) + PT(J) * (P(J+2,I) - P(J,I))                   REFINE  94
  175 CONTINUE                                                          REFINE  95
      DO 180 J=JST,JM2,2                                                REFINE  96
      P(J+1,I) = P(J,I) + PT(J) * (P(J+2,I) - P(J,I))                   REFINE  97
  180 CONTINUE                                                          REFINE  98
  190 CONTINUE                                                          REFINE  99
C             USE EXTRAPOLATION FOR JLOW,JUP                            REFINE 100
      D1 = Y(JLOW) - Y(JLOW-1)                                          REFINE 101
      D2 = Y(JLOW-1) - Y(JLOW-2)                                        REFINE 102
      CL1 = (D1 + D2) / D2                                              REFINE 103
      CL2 = D1/D2                                                       REFINE 104
      D1 = Y(JUP+1) - Y(JUP)                                            REFINE 105
      D2 = Y(JUP+2) - Y(JUP+1)                                          REFINE 106
      CU1 = (D1 + D2) / D2                                              REFINE 107
      CU2 = D1 / D2                                                     REFINE 108
      DO 200 I = IMIN,IMAX                                              REFINE 109
      P(JUP,I) = CU1*P(JUP+1,I) - CU2*P(JUP+2,I)                        REFINE 110
      P(JLOW,I) = CL1*P(JLOW-1,I) - CL2*P(JLOW-2,I)                     REFINE 111
  200 CONTINUE                                                          REFINE 112
      RETURN                                                            REFINE 113
      END                                                               REFINE 114
```

```
      SUBROUTINE RESET                                                  RESET    2
C             SUBROUTINE RESET  UPDATES FAR FIELD BOUNDARY              RESET    3
C             CONDITIONS FOR SUBSONIC FREESTREAM FLOWS.                 RESET    4
C             CALLED BY - SOLVE.                                        RESET    5
                                                                        RESET    6
      COMMON        P(102,101),X(100)  , Y(100)                         BLANK    2
      COMMON / COM1/ IMIN     , IMAX    , IUP     , IDOWN   , ILE    ,  COM1     2
     1              ITE      , JMIN    , JMAX    , JUP     , JLOW   ,  COM1     3
     2              JTOP     , JBOT                                     COM1     4
      COMMON / COM2/ AK       , ALPHA   , DUB     , GAM1    , RTK       COM2     2
      LOGICAL       ABORT                                               COM3     2
      COMMON / COM3/ IREF     , ABORT   , ICUT    , KSTEP               COM3     3
      COMMON /COM24/ DTOP(100), DBOT(100),DUP(100), DDOWN(100),         COM24    2
     1              VTOP(100), VBOT(100),VUP(100), VDOWN(100)          COM24    3
      INTEGER       BCTYPE                                              COM28    2
      COMMON /COM28/ BCTYPE   , CIRCFF  , FHINV   , POR     , CIRCTE    COM28    3
                                                                        RESET   13
C             SET BOUNDARY CONDITIONS AT UPSTREAM AND DOWNSTREAM        RESET   14
C             ENDS.                                                     RESET   15
      K = JMIN - KSTEP                                                  RESET   16
      DO 10 J = JMIN,JMAX                                               RESET   17
      K = K + KSTEP                                                     RESET   18
      IF (J .EQ. JUP) K = K + KSTEP - 1                                 RESET   19
      P(J,IMIN) = CIRCFF*VUP(K) + DUB*DUP(K)                            RESET   20
      P(J,IMAX) = CIRCFF*VDOWN(K) + DUB*DDOWN(K)                        RESET   21
 10   CONTINUE                                                          RESET   22
      IF(BCTYPE .NE. 1) GO TO 25                                        RESET   23
C             UPDATE BOUNDARY CONDITIONS ON TOP AND BOTTOM              RESET   24
      K = IMIN - KSTEP                                                  RESET   25
      DO 20 I = IMIN,IMAX                                               RESET   26
      K = K + KSTEP                                                     RESET   27
      P(JMIN,I) = CIRCFF*VBOT(K) + DUB*DBOT(K)                          RESET   28
      P(JMAX,I) = CIRCFF*VTOP(K) + DUB*DTOP(K)                          RESET   29
 20   CONTINUE                                                          RESET   30
 25   RETURN                                                            RESET   31
      END                                                               RESET   32
      SUBROUTINE SAVEP
C                          SAVEP MOVES DATA INTO OLD DATA LOCATIONS AND
C                          WRITES IT ON TAPE3 IF REQUESTED.
C                          CALLED BY - TSFOIL.
      COMMON          P(102,101),X(100)  , Y(100)
      COMMON / COM1/ IMIN     , IMAX   , IUP    , IDOWN  , ILE    ,
     1               ITE      , JMIN   , JMAX   , JUP    , JLOW   ,
     2               JTOP     , JBOT
      COMMON / COM2/ AK       , ALPHA  , DUB    , GAM1   , RTK
      LOGICAL        ABORT
      COMMON / COM3/ IREF     , ABORT  , ICUT   , KSTEP
      LOGICAL        AMESH
      COMMON / COM4/ XIN(100) , YIN(100), AMESH
      COMMON / COM6/ FL(100)  , FXL(100), FU(100) , FXU(100),
     1          CAMBER(100), THICK(100),VOL     , XFOIL(100), IFOIL
      INTEGER        PSTART
      LOGICAL        PSAVE
      COMMON /COM11/ ALPHA0   , CLOLD  , DELTA0 , DUB0   , EMACH0 ,
     1               IMIN0    , IMAX0  , IMAXI  , JMIN0  , JMAX0  ,
     2               JMAXI    , PSAVE  , PSTART , TITLE(20), TITLEO(20),
     3               VOL0     , XOLD(100),YOLD(100)
      COMMON /COM12/ F        , H      , HALFPI , PI     , RTKPOR ,
     1               TWOPI
      LOGICAL        PHYS
      INTEGER        PRTFLO   , SIMDEF
      COMMON /COM27/ CL       , DELTA  , DELRT2 , EMACH  , EMROOT ,
     1               PHYS     , PRTFLO , SIMDEF , SONVEL , VFACT  ,
     2               YFACT
      INTEGER        BCTYPE
      COMMON /COM28/ BCTYPE   , CIRCFF , FHINV  , POR    , CIRCTE
C                          RESET PARAMETERS SCALED IN SUBROUTINE SCALE.
      ALPHA = ALPHA * VFACT
      H = H * YFACT
      POR = POR / YFACT
      DO 6 J=JMIN,JMAX
      YIN(J) = YIN(J) * YFACT
    6 CONTINUE
      IF (ABORT) RETURN
C                          MOVE RESTART DATA TO OLD BLOCK.
      DO 10 I=1,20
      TITLEO(I) = TITLE(I)
   10 CONTINUE
      IMIN0 = IMIN
      JMIN0 = JMIN
      IMAX0  = IMAX
      JMAX0  = JMAX
      CLOLD  = CL
      EMACH0 = EMACH
      ALPHA0 = ALPHA
      DELTA0 = DELTA
      VOL0   = VOL
      DUB0 = DUB

      DO 20 I=IMIN0,IMAX0
      XOLD(I) = X(I)
   20 CONTINUE

      DO 30 J=JMIN0,JMAX0
      YOLD(J) = Y(J)
   30 CONTINUE
C                          CHECK TO SEE IF RESTART IS TO BE WRITTEN ON
C                                 TAPE3.
      IF (.NOT. PSAVE) GO TO 100
   40 CONTINUE
      WRITE (3,900) TITLEO
      WRITE (3,901) IMAX0, JMAX0, IMIN0, JMIN0
      WRITE (3,902) CLOLD , EMACH0, ALPHA0, DELTA0, VOL0, DUB0
      WRITE (3,902) (XOLD(I),I=IMIN0,IMAX0)
      WRITE (3,902) (YOLD(J),J=JMIN0,JMAX0)
      DO 50 I=IMIN0,IMAX0
      WRITE (3,902) (P(J,I),J=JMIN0,JMAX0)
   50 CONTINUE
  100 CONTINUE
      RETURN
  900 FORMAT(20A4)
  901 FORMAT(4I5)
  902 FORMAT(8F10.6)

      END
```

```
      SUBROUTINE SCALE
C                     SUBROUTINE SCALES PHYSICAL VARIABLES TO TRANSONIC
C                     VARIABLES.
C                     IF PHYS = .TRUE., ALL INPUT/OUTPUT QUANTITIES ARE IN
C                     PHYSICAL UNITS NORMALIZED BY FREESTREAM VALUES AND
C                     AIRFOIL CHORD. THIS SUBROUTINE THEN SCALES THE
C                     QUANTITIES TO TRANSONIC VARIABLES BY THE FOLLOWING
C                     CONVENTION
C                          SIMDEF = 1  COLE SCALING
C                          SIMDEF = 2  SPREITER SCALING
C                          SIMDEF = 3  KRUPP SCALING
C                          SIMDEF = 4  USER CHOICE
C                     IF PHYS = .FALSE., INPUT IS ALREADY IN SCALED
C                     VARIABLES AND NO FURTHER SCALING IS DONE.
C                     CALLED BY - TSFOIL.
      COMMON          P(102,101),X(100)  , Y(100)
      COMMON / COM1/ IMIN     , IMAX   , IUP    , IDOWN  , ILE    ,
     1               ITE      , JMIN   , JMAX   , JUP    , JLOW   ,
     2               JTOP     , JBOT
      COMMON / COM2/ AK       , ALPHA  , DUB    , GAM1   , RTK
      LOGICAL        AMESH
      COMMON / COM4/ XIN(100) , YIN(100), AMESH
      INTEGER        PSTART
      LOGICAL        PSAVE
      COMMON /COM11/ ALPHA0   , CLOLD  , DELTA0 , DUB0   , EMACH0 ,
     1               IMIN0    , IMAX0  , IMAXI  , JMIN0  , JMAX0  ,
     2               JMAXI    , PSAVE  , PSTART , TITLE(20), TITLEO(20),
     3               VOL0     , XOLD(100),YOLD(100)
      COMMON /COM12/ F        , H      , HALFPI , PI     , RTKPOR ,
     1               TWOPI
      COMMON /COM13/ CDFACT   , CLFACT , CMFACT , CPFACT , CPSTAR
      LOGICAL        PHYS
      INTEGER        PRTFLO   , SIMDEF
      COMMON /COM27/ CL       , DELTA  , DELRT2 , EMACH  , EMROOT ,
     1               PHYS     , PRTFLO , SIMDEF , SONVEL , VFACT  ,
     2               YFACT
      INTEGER        BCTYPE
      COMMON /COM28/ BCTYPE   , CIRCFF , FHINV  , POR    , CIRCTE

      IF(PHYS) GO TO 50
C                     PHYS = .FALSE.  NO SCALING
      CPFACT = 1.
      CDFACT = 1.
      CLFACT = 1.
      CMFACT = 1.
      YFACT = 1.
      VFACT = 1.
      GO TO 400
C                     PHYS = .TRUE.  COMPUTE CONSTANTS
   50 CONTINUE
      EMACH2 = EMACH*EMACH
      BETA  = 1. - EMACH2
      DELRT1 = DELTA**(1./3.)
      DELRT2 = DELTA**(2./3.)
C
C                     BRANCH TO APPROPRIATE SCALING
      GO TO (100,200,300,400), SIMDEF
C
C                     SIMDEF = 1
C                     COLE SCALING
  100 CONTINUE
      AK = BETA/DELRT2
      YFACT = 1./DELRT1
      CPFACT = DELRT2
      CLFACT = DELRT2
      CDFACT = DELRT2*DELTA
      CMFACT = DELRT2
      VFACT = DELTA*57.295779
      GO TO 500
C                     SIMDEF = 2
C                     SPREITER SCALING
  200 CONTINUE
      EMROOT = EMACH**(2./3.)
      AK = BETA/(DELRT2*EMROOT*EMROOT)
      YFACT = 1./(DELRT1*EMROOT)
      CPFACT = DELRT2/EMROOT
      CLFACT = CPFACT
      CMFACT = CPFACT
      CDFACT = CPFACT*DELTA
      VFACT = DELTA*57.295779
      GO TO 500
C                     SIMDEF = 3
C                     KRUPP SCALING
```

```
  500   CONTINUE                                                              SCALE    73
        AK = BETA/(DELRT2*EMACH)                                              SCALE    74
        YFACT = 1./(DELRT1*EMACH**.5)                                         SCALE    75
        CPFACT = DELRT2/(EMACH**.75)                                          SCALE    76
        CLFACT = CPFACT                                                       SCALE    77
        CMFACT = CPFACT                                                       SCALE    78
        CDFACT = CPFACT*DELTA                                                 SCALE    79
        VFACT = DELTA*57.295779                                               SCALE    80
        GO TO 500                                                             SCALE    81
  400   CONTINUE                                                              SCALE    82
C                     SIMDEF = 4                                              SCALE    83
C                     THIS ADDRESS IS INACTIVE                                SCALE    84
C                     USER MAY INSERT SCALING OF OWN CHOICE                   SCALE    85
C                     DEFINITION FOR LOCAL MACH NUMBER MUST BE ADJUSTED       SCALE    86
C                     IN EMACH1.                                              SCALE    87
        WRITE(6,1000)                                                         SCALE    88
 1000 FORMAT(34H1ABNORMAL STOP IN SUBROUTINE SCALE/                           SCALE    89
     1       24H SIMDEF=4 IS NOT USEABLE)                                     SCALE    90
        STOP                                                                  SCALE    91
  500   CONTINUE                                                              SCALE    92
C                     SCALE Y MESH                                            SCALE    93
        YFACIV = 1.0 / YFACT                                                  SCALE    94
        DO 502 J=JMIN,JMAX                                                    SCALE    95
        YIN(J)  = YIN(J)  * YFACIV                                            SCALE    96
  502  CONTINUE                                                               SCALE    97
        IF (PSTART .EQ. 1) GO TO 505                                          SCALE    98
        DO 504 J=JMINO,JMAXO                                                  SCALE    99
        YOLD(J) = YOLD(J) * YFACIV                                            SCALE   100
   504 CONTINUE                                                               SCALE   101
   505 CONTINUE                                                               SCALE   102
C                     SCALE TUNNEL PARAMETERS                                 SCALE   103
        H = H/YFACT                                                           SCALE   104
        POR = POR*YFACT                                                       SCALE   105
                                                                              SCALE   106
C                     SCALE ANGLE OF ATTACK                                   SCALE   107
        ALPHA = ALPHA/VFACT                                                   SCALE   108
                                                                              SCALE   109
  600   CONTINUE                                                              SCALE   110
C                     CHECK VALUE OF AK FOR DEFAULT.                          SCALE   111
        IF (AK .EQ. 0.0) CALL INPERR (7)                                      SCALE   112
C                     COMPUTE SQUARE ROOT OF AK                               SCALE   113
        RTK = SQRT(ABS(AK))                                                   SCALE   114
C                     COMPUTE SONIC VELOCITY                                  SCALE   115
        IF (ABS(GAM1).GT..0001) GO TO 999                                     SCALE   116
        SONVEL=1.                                                             SCALE   117
        CPSTAR=0.                                                             SCALE   118
        RETURN                                                                SCALE   119
   999 CONTINUE                                                               SCALE   120
        SONVEL = AK / GAM1                                                    SCALE   121
        CPSTAR = -2.0 * SONVEL * CPFACT                                       SCALE   122
        RETURN                                                                SCALE   123
        END                                                                   SCALE   124
```

```
      SUBROUTINE SETBC                                                        SETBC     2
C                   SUBROUTINE SETBC SETS THE LIMITS ON RANGE OF I AND J      SETBC     3
C                   FOR SOLUTION OF THE DIFFERENCE EQUATIONS.                 SETBC     4
C                   THE BODY SLOPE BOUNDARY CONDITION  AT THE CURRENT         SETBC     5
C                   X MESH POINTS ON THE BODY ARE MULTIPLIED BY MESH          SETBC     6
C                   SPACING CONSTANTS AND ENTERED INTO ARRAYS FXUBC AND       SETBC     7
C                   FXLBC FOR USE IN SUBROUTINE SYOR.                         SETBC     8
C                   CALLED BY - TSFOIL.                                       SETBC     9
                                                                              SETBC    10
      COMMON          P(102,101),X(100)  , Y(100)                             BLANK     2
      COMMON / COM1/ IMIN     , IMAX    , IUP     , IDOWN    , ILE     ,      COM1      2
     1               ITE      , JMIN    , JMAX    , JUP      , JLOW    ,      COM1      3
     2               JTOP     , JBOT                                          COM1      4
      COMMON / COM2/ AK       , ALPHA   , DUB     , GAM1     , RTK            COM2      2
      LOGICAL        ABORT                                                    COM3      2
      COMMON / COM3/ IREF     , ABORT   , ICUT    , KSTEP                     COM3      3
      COMMON / COM6/ FL(100)  , FXL(100), FU(100) , FXU(100),                 COM6      2
     1             CAMBER(100), THICK(100),VOL    , XFOIL(100), IFOIL         COM6      3
      COMMON /COM17/ CYYBLC   , CYYBLD  , CYYBLU  , CYYBUC   , CYYBUD  ,      COM17     2
     1               CYYBUU   ,FXLBC(100),FXUBC(100)                          COM17     3
      INTEGER        BCTYPE                                                   COM20     2
      COMMON /COM20/ BCTYPE   , CIRCFF  , FHINV   , POR      , CIRCTL         COM20     3
                                                                              SETBC    18
C                   SET LIMITS ON I AND J INDICIES                            SETBC    19
      INT = 0                                                                 SETBC    20
      IF(AK .LT. 0.) INT = 1                                                  SETBC    21
      IUP = IMIN + 1 + INT                                                    SETBC    22
      IDOWN = IMAX - 1 + INT                                                  SETBC    23
      JINT = 0                                                                SETBC    24
      IF(BCTYPE .EQ. 1 .AND. AK .GT. 0.) JINT = 1                             SETBC    25
      IF(BCTYPE .EQ. 3) JINT = 1                                              SETBC    26
      IF(BCTYPE .EQ. 5 .AND. POR .GT. 1.5) JINT = 1                           SETBC    27
      JBOT = JMIN + JINT                                                      SETBC    28
      JTOP = JMAX - JINT                                                      SETBC    29
C                   AIRFOIL BODY BOUNDARY CONDITION                           SETBC    30
C                   ZERO ELEMENTS IN ARRAYS FOR UPPER AND LOWER BODY          SETBC    31
C                   BOUNDARY CONDITIONS                                       SETBC    32
      DO 30 I = IMIN,IMAX                                                     SETBC    33
      FXLBC(I) = 0.                                                           SETBC    34
      FXUBC(I) = 0.                                                           SETBC    35
   30 CONTINUE                                                                SETBC    36
C                   ENTER BODY SLOPES AT MESH POINTS ON AIRFOIL               SETBC    37
C                   INTO ARRAYS FOR BODY BOUNDARY CONDITIONS                  SETBC    38
      IF(IREF .LE. 0) KSTEP = 1                                               SETBC    39
      IF(IREF .EQ. 1)  KSTEP = 2                                              SETBC    40
      IF(IREF .EQ. 2)  KSTEP = 4                                              SETBC    41
      NFOIL = ITE - ILE + 1                                                   SETBC    42
      IF = IFOIL + KSTEP                                                      SETBC    43
      I = ITE + 1                                                             SETBC    44
      DO 50 N = 1,NFOIL                                                       SETBC    45
      I = I-1                                                                 SETBC    46
      IF = IF - KSTEP                                                         SETBC    47
      FXLBC(I) = CYYBLU*(FXL(IF) - ALPHA)                                     SETBC    48
      FXUBC(I) = CYYBUD*(FXU(IF) - ALPHA)                                     SETBC    49
   50 CONTINUE                                                                SETBC    50
      RETURN                                                                  SETBC    51
      END                                                                     SETBC    52
```

```
      SUBROUTINE SIMP(R,X,Y,N,IER)                                            SIMP      2
C                   SUBROUTINE TO INTEGRATE BY SIMPSONS RULE.                 SIMP      3
C                   CALLED BY BODY.                                           SIMP      4
      DIMENSION X(N),Y(N)                                                     SIMP      5
      R=0.0                                                                   SIMP      6
      IF(N.GT.1) GO TO 1                                                      SIMP      7
      IER=2                                                                   SIMP      8
      RETURN                                                                  SIMP      9
    1 IF(X(1).EQ.X(2)) GO TO 12                                               SIMP     10
      NM1=N-1                                                                 SIMP     11
      IF(N.EQ.2) GO TO 13                                                     SIMP     12
      IF(X(1).LT.X(2)) GO TO 3                                                SIMP     13
C   TEST FOR X TO BE MONOTONICALLY DECREASING                                 SIMP     14
      DO 2 I=2,NM1                                                            SIMP     15
      IF(X(I+1).GE.X(I)) GO TO 12                                             SIMP     16
    2 CONTINUE                                                                SIMP     17
      GO TO 5                                                                 SIMP     18
C   TEST FOR X TO BE MONOTONICALLY INCREASING                                 SIMP     19
    3 DO 4 I=2,NM1                                                            SIMP     20
      IF(X(I+1).LE.X(I)) GO TO 12                                             SIMP     21
    4 CONTINUE                                                                SIMP     22
    5 NM2=N-2                                                                 SIMP     23
      IF(MOD(N,2).EQ.0) GO TO 14                                              SIMP     24
      P=0.0                                                                   SIMP     25
      N1=1                                                                    SIMP     26
    6 S1=X(N1+1)-X(N1)                                                        SIMP     27
      S2=X(N1+2)-X(N1+1)                                                      SIMP     28
      S3=X(NM1)-X(NM2)                                                        SIMP     29
      S4=X(N)-X(NM1)                                                          SIMP     30
      P=(2.*S1**2+S1*S2-S2**2)/S1*Y(N1)+(2.*S4**2+S3*S4-S3**2)/S4*Y(N)        SIMP     31
      N1=N1+1                                                                 SIMP     32
      DO 7 I=N1,NM1,2                                                         SIMP     33
      S1=X(I)-X(I-1)                                                          SIMP     34
      S2=X(I+1)-X(I)                                                          SIMP     35
    7 R=R+(S1+S2)**3/(S1*S2)*Y(I)                                             SIMP     36
      IF(N.LT.5) GO TO 9                                                      SIMP     37
      N1=N1+1                                                                 SIMP     38
      DO 8 I=N1,NM2,2                                                         SIMP     39
      S1=X(I-1)-X(I-2)                                                        SIMP     40
      S2=X(I)-X(I-1)                                                          SIMP     41
      S3=X(I+1)-X(I)                                                          SIMP     42
      S4=X(I+2)-X(I+1)                                                        SIMP     43
    8 R=R+((2.*S2**2+S1*S2-S1**2)/S2+(2.*S3**2+S3*S4-S4**2)/S3)*Y(I)          SIMP     44
    9 R=R/6.+P                                                                SIMP     45
   10 CONTINUE                                                                SIMP     46
      IER=1                                                                   SIMP     47
      RETURN                                                                  SIMP     48
   12 IER=4
      RETURN
C  TRAPEZOIDAL RULE FOR N=2
   13 P=(X(2)-X(1))*(Y(1)+Y(2))/2.0
      GO TO 10
C  FIT POLYNOMIAL THRU FIRST 3 POINTS AND INTEGRATE FROM X(1) TO X(2).
   14 S1=X(2)-X(1)
      S2=X(3)-X(1)
      S3=Y(2)-Y(1)
      S4=Y(3)-Y(1)
      P=S1/6.*(2.*S3+6.*Y(1)+(S2**2*S3-S1**2*S4)/(S2*(S2-S1)))
      N1=2
      GO TO 6
      END
```

```
      SUBROUTINE SOLVE

C                   SOLVE CONTROLS THE MAIN ITERATION LOOP.
C                   CALLED BY - TSFOIL.
      COMMON          P(102,101),X(100)   , Y(100)
      COMMON / COM1/ IMIN     , IMAX     , IUP      , IDOWN    , ILE      ,
     1               ITE      , JMIN     , JMAX     , JUP      , JLOW     ,
     2               JTOP     , JBOT
      COMMON / COM2/ AK       , ALPHA    , DUB      , GAM1     , RTK
      LOGICAL         ABORT
      COMMON / COM3/ IREF     , ABORT    , ICUT     , KSTEP
      COMMON / COM7/ CJUP     , CJUP1    , CJLOW    , CJLOW1
      COMMON / COM8/ CVERGE   , DVERGE   , IPRTER   , MAXIT    ,
     1               WE(3)    , EPS
      COMMON /COM13/ CDFACT   , CLFACT   , CMFACT   , CPFACT   , CPSTAR
      LOGICAL         OUTERR
      COMMON /COM18/ ERROR    , I1       , I2       , IERROR   , JERROR   ,
     1               OUTERR   , EMU(100,2)          , VC(100)  ,
     2               WI       , DCIRC    , POLD(100,2)
      COMMON /COM22/ CXC(100) , CXL(100), CXR(100), CXXC(100),CXXL(100),
     1               CXXR(100), C1(100)
      COMMON /COM25/ CPL(100) , CPU(100)
      INTEGER         BCTYPE
      COMMON /COM28/ BCTYPE   , CIRCFF   , FHINV    , POR      , CIRCTE
      COMMON /COM32/ BIGRL    , IRL      , JRL
      COMMON /COM33/ THETA(100,100)

      REAL LIFT
      DATA    NDUB / 25 /
C
C
      ABORT = .FALSE.
      WRITE(6,600)
C
      IF (IREF .EQ. 2) MAXITM = MAXIT / 4
      IF (IREF .EQ. 1) MAXITM = MAXIT / 2
      IF (IREF .EQ. 0) MAXITM = MAXIT
      KK = 3 - IREF
      IF (KK .EQ. 1) WRITE(6,607)
      IF (KK .EQ. 2) WRITE(6,608)
      IF (KK .EQ. 3) WRITE(6,609)
      WEP = WE(KK)
      WI = 1.0 / WEP
      WRITE (6,606) WEP, EPS, MAXITM
C
      WRITE(6,601)

      DO 1 ITER = 1,MAXITM
C
C                   INITIALIZE EMU
      I1=1
      I2=2
      DO 5 J=JMIN,JMAX
      POLD(J,I2) = P(J,IUP-1)
      EMU(J,I2) = 0.0
    5 CONTINUE

      IF ( AK .GT. 0.0 ) GO TO 7
      DO 6 J=JMIN,JMAX
      EMU(J,I2) = C1(2)
    6 CONTINUE
    7 CONTINUE
      OUTERR=.FALSE.
      IF(MOD(ITER,IPRTER) .EQ. 0)  OUTERR=.TRUE.
      ERROR=0.0
      BIGRL = 0.0
C                         UPDATE PJUMP.
      CALL RECIRC

      CALL SYOR

C                         UPDATE CIRCULATION FOR SUBSONIC FREESTREAM FLOW

      IF ( AK .LT. 0.0 ) GO TO 10
      IF (BCTYPE .NE. 1) GO TO 9
      IK = IUP - IMIN
      DO 8 I=IUP,IDOWN
      IK = IK + KSTEP
      JK = JBOT - JMIN
      DO 82 J=JBOT,JTOP
      JINC = KSTEP
      IF (Y(J) .LT. 0.0 .AND. Y(J+1) .GT. 0.0) JINC = 2 * KSTEP - 1
      JK = JK + JINC
      P(J,I) = P(J,I) + DCIRC * THETA(JK,IK)
   82 CONTINUE
    8 CONTINUE
    9 CONTINUE
C*******
      IF (MOD(ITER,NDUB) .EQ. 0) CALL REDUB
C*******
      CALL RESET
   10 CONTINUE
      IF(OUTERR)  GO TO 2
      GO TO 1
 2    CONTINUE
      CL = LIFT (CLFACT)
      CM = PITCH (CMFACT)
      ERCIRC=ABS(DCIRC)
      CPMAXL=0.0
      CPMAXU=0.0
      DO 14 I=ILE,ITE
      UL=CJLOW*PX(I,JLOW)-CJLOW1*PX(I,JLOW-1)
      CPOLD=CPL(I)
      CPL(I)=-2.0*UL*CPFACT
      CPERR=ABS(CPL(I)-CPOLD)
      IF (CPERR .LE. CPMAXL) GO TO 12
      CPMAXL=CPERR
      IERRL=I
   12 UU=CJUP*PX(I,JUP)-CJUP1*PX(I,JUP+1)
       CPOLD=CPU(I)
      CPU(I)=-2.0*UU*CPFACT
      CPERR=ABS(CPU(I)-CPOLD)
      IF (CPERR .LE. CPMAXU) GO TO 14
      CPMAXU=CPERR

      IERRU=I
   14 CONTINUE
      WRITE (6,602) ITER, CL, CM, IERROR, JERROR, ERROR, IRL, JRL,
     1 BIGRL,ERCIRC,IERRU,CPMAXU,IERRL,CPMAXL
      IF (ERROR .LE. CVERGE) GO TO 3
      IF (ERROR .GE. DVERGE) GO TO 4
1     CONTINUE
      WRITE(6,605)
      RETURN
3     CONTINUE
      WRITE(6,603)
      RETURN
4     CONTINUE
      ABORT = .TRUE.
      WRITE(6,604)
      RETURN

600   FORMAT(1H1)
 601  FORMAT(11H0 ITERATION,4X,2HCL,9X,2HCM,5X,4HIERR,2X,4HJERR,5X,
    1     5HERROR,5X,3HIRL,3X,3HJRL,6X,5HBIGRL,8X,6HERCIRC,4X,4HICPU,
    2     5X,6HCPERRU,4X,4HICPL,5X,6HCPERRL/)
 602  FORMAT(3X,I4,1X,2F11.5,2I6,E13.4,2I6,2E13.4,2(I6,E13.4))
 603  FORMAT(//20X,34H*******  SOLUTION CONVERGED  ******)
604   FORMAT(//20X,33H*******  SOLUTION DIVERGED  ******)
605   FORMAT(//20X,39H*******  ITERATION LIMIT REACHED  ******)
 606  FORMAT(10X,5HWE = F8.4,5X,6HEPS = F8.4,5X,
    1      22HMAXIT FOR THIS MESH = I4)
 607  FORMAT(40X,33HINTERMEDIATE OUTPUT FOR COARSE MESH//)
 608  FORMAT(40X,33HINTERMEDIATE OUTPUT FOR MEDIUM MESH//)
 609  FORMAT(40X,33HINTERMEDIATE OUTPUT FOR FINE MESH//)

      END
```

Microfilmed From
Best Available Copy

```
      SUBROUTINE SPLN1 (X,Y,N)
C                     CALLED BY - BODY.
      COMMON /SPLN/ A(200)     , B(200)  , DY1     , DY2     , K1
     1              K2         , XP      , YP      ,DYP

CONTINUOUS DERIVATIVE INTERPOLATION SUBROUTINES
CURFIT COMPUTES COEFFICIENTS OF CUBICS  --  A(I)...I=1,2*N-2
C.....FOR THE WHOLE TABULATED TABLE
C     X(I) = INDEPENDENT VARIABLE.....I=1,N  (GIVEN)
C     Y(I) = DEPENDENT VARIABLE.....I=1,N  (GIVEN)
C     N = LENGTH OF Y-VS-X TABLE  (GIVEN)
C     DY1 = 1ST OR 2ND DERIVATIVE AT LOWER END OF TABLE
C     DY2 = 1ST OR 2ND DERIVATIVE AT UPPER END OF TABLE
C     K1 = 1 ......DY1 = 1ST DERIVATIVE  (GIVEN)
C     K1 = 2 ......DY1 = 2ND DERIVATIVE  (GIVEN)
C     K2 = 1 ......DY2 = 1ST DERIVATIVE  (GIVEN)
C     K2 = 2 ......DY2 = 2ND DERIVATIVE  (GIVEN)

      DIMENSION  X(1),  Y(1)

      N1=N-2
      C1=X(2)-X(1)
      IF (K1 .EQ. 2) GO TO 4
      B(1) = 0.
      A(1)=(DY1-(Y(2)-Y(1))/C1)/C1
      GO TO 5
    4 B(1)=-C1
      A(1)=-DY1/2.
5     J=1
      IF(N.EQ.2)GO TO 42
      DO 10 I=1,N1
      J=J+1
      C1=X(I+1)-X(I)
      C2=X(I+2)-X(I+1)
      C3=Y(I+1)-Y(I)
      C4=Y(I+2)-Y(I+1)
      C5=C3/C1-C4/C2
      C6=C1/C2
      C7=C1*C2
      B(J)=1.0/(C6*(C1-B(J-1)))
      A(J)=(C5/C2-C6*A(J-1))*B(J)
      J=J+1
      B(J)=1.0/((-C1-C2)/C7-C6*B(J-1))
      A(J) =(-C5/C7-C6*A(J-1))*B(J)
10    CONTINUE
      IF(K2.EQ.2)GO TO 30
      A(J+1) = (DY2-C4/C2+C2*A(J))/(C2*(B(J)-C2))
      GO TO 45
30    A(J+1)=(DY2/2.+A(J))/(-2.*C2+B(J))
      GO TO 45
C     STATEMENTS 42 TO 44 ARE FOR N=2 ONLY
42    C3=K1
      C2=1.0/C3
      IF(K2.EQ.2)GO TO 44
      A(J+1) = ((Y(2)-Y(1))/C1-A(J)*C1-DY2)/(C1*C1)*C2
      GO TO 45
44    A(J+1)=C3*((DY2+2.0*A(1))/(4.0*C1))
45    J=2*(N-1)
50    J=J-1
      IF(J.LE.0)RETURN
      A(J) = A(J)-B(J)*A(J+1)
      GO TO 50
C
C   ENTRY POINT FOR INTERPOLATION
      ENTRY SPLN1X

      IF(XP.GT.X(1)) GO TO 11
C     SPECIAL CASE FOR EXTRAPOLATION BEYOND LOWER END OF X-TABLE
      C=X(2)-X(1)
      DYP=(Y(2)-Y(1))/C+A(1)*C
      YP=Y(1)+DYP*(XP-X(1))
      RETURN
   11 IF(XP.LT.X(N)) GO TO 13
C     SPECIAL CASE FOR EXTRAPOLATION BEYOND UPPER END OF Y-TABLE
      C=X(N)-X(N-1)
      DYP=(Y(N)-Y(N-1))/C-A(2*N-3)*C-A(2*N-2)*C*C
      YP=Y(N)+DYP*(XP-X(N))
      RETURN
   13 I=1
   14 I=I+1
      IF(X(I).LT.XP) GO TO 14
C     NOW XP HAS BEEN BRACKETED SO THAT X(I-1).LT.XP.LE.X(I)
      C=XP-X(I-1)
      D=X(I)-XP
      K=2*I-3
      SLOPE=(Y(I)-Y(I-1))/(X(I)-X(I-1))
      YP=Y(I-1)+(SLOPE+(A(K)+A(K+1)*C)*D)*C
               DYP=SLOPE+A(K)*(D-C)+A(K+1)*(2.*D-C)*C
       RETURN
      END
```

```
      SUBROUTINE SYOR

C                   SYOR COMPUTES NEW P AT ALL MESH POINTS.
C                   CALLED BY - SOLVE.
      COMMON        P(102,101),X(100)  , Y(100)
      COMMON / COM1/ IMIN      , IMAX    , IUP     , IDOWN   , ILE     ,
     1              ITE        , JMIN    , JMAX    , JUP     , JLOW    ,
     2              JTOP       , JBOT
      COMMON / COM2/ AK        , ALPHA   , DUB     , GAM1    , RTK
      COMMON / COM8/ CVERGE    , DVERGE  , IPRTER  , MAXIT   ,
     1              WE(3)      , EPS
      LOGICAL       FCR        , KUTTA
      COMMON /COM14/ CLSET     , FCR     , KUTTA   , WCIRC
      COMMON /COM17/ CYYBLC    , CYYBLD  , CYYBLU  , CYYBUC  , CYYBUD  ,
     1              CYYBUU     ,FXLBC(100),FXUBC(100)
      LOGICAL       OUTERR
      COMMON /COM18/ ERROR     , I1      , I2      , IERROR  , JERROR  ,
     1              OUTERR     , EMU(100,2)        , VC(100) ,
     2              WI         , DCIRC   , POLD(100,2)
      COMMON /COM19/ DIAG(100), RHS(100), SUB(100), SUP(100)
      COMMON /COM22/ CXC(100) , CXL(100), CXR(100), CXXC(100),CXXL(100),
     1              CXXR(100), C1(100)
      COMMON /COM23/ CYYC(100), CYYD(100),CYYU(100), IVAL
      COMMON /COM24/ PJUMP(100)
      COMMON /COM32/ SIGRL     , IRL     , JRL
      DIMENSION      SAVE(100)

      IM2=IUP-1
      IF( AK .LT. 0.0 )  IM2=IUP-2


      J1 = JBOT + 1
      J2 = JTOP - JBOT

      DO 200 I=IUP,IDOWN
      EPSX = EPS/((X(I)-X(I-1))**2)

C                           COMPUTE VC = 1 - M**2

      DO 10 J=JBOT,JTOP
      VC(J) = C1(I) - (CXL(I)*POLD(J,I2) + CXC(I)*P(J,I)
     1             +  CXR(I)*P(J,I+1))
      EMU(J,I1) = 0.0
      POLD(J,I1) = P(J,I)
   10 CONTINUE
      DO 20 J=JBOT,JTOP
      IF (VC(J) .LT. 0.0) EMU(J,I1) = VC(J)
   20 CONTINUE
      IF ( FCR ) GO TO 22
      DO 21 J=JBOT,JTOP
      EMU(J,I2) = EMU(J,I1)
   21 CONTINUE
   22 CONTINUE

C                           COMPUTE ELEMENTS OF MATRIX

      DO 30 J=JBOT,JTOP
      DIAG(J) = (EMU(J,I1) - VC(J)) * CXXC(I) + WI
     1           + EMU(J,I2) * CXXR(I-1) - CYYC(J)
      SUP(J)=CYYD(J)
      SUB(J)=CYYU(J)
   30 CONTINUE

C                           COMPUTE RESIDUAL

      DO 40 J=JBOT,JTOP
      RHS(J)    = -(VC(J)-EMU(J,I1))*
     1          (CXXL(I)*P(J,I-1) - CXXC(I)*P(J,I) + CXXR(I)*P(J,I+1))
   40 CONTINUE
      DO 50 J=JBOT,JTOP
      RHS(J)    = RHS(J) - (EMU(J,I2) * (CXXL(I-1)*P(J,IM2)
```

176

```
     1            - CXXC(I-1)*P(J,I-1) + CXXR(I-1)*P(J,I)))
   50 CONTINUE
      JA = JBOT + 1
      JB = JTOP - 1
      DO 60 J=JA,JB
      RHS(J)   = RHS(J) - (CYYD(J)*P(J-1,I) - CYYC(J)*P(J,I)
     1                                  + CYYU(J)*P(J+1,I))
   60 CONTINUE
      RHS(JBOT) = RHS(JBOT) - (-CYYC(JBOT)*P(JBOT,I)
     1                     + CYYU(JBOT)*P(JBOT+1,I))
      IF (JBOT .EQ. JMIN) GO TO 61
      RHS(JBOT) = RHS(JBOT) -CYYD(JBOT)*P(JBOT-1,I)
   61 CONTINUE
      RHS(JTOP) = RHS(JTOP) - (CYYD(JTOP)*P(JTOP-1,I)
     1                     - CYYC(JTOP)*P(JTOP,I))
      IF (JTOP .EQ. JMAX) GO TO 62
      RHS(JTOP) = RHS(JTOP) - CYYU(JTOP)*P(JTOP+1,I)
   62 CONTINUE

C                      CHECK FOR AIRFOIL B.C. AND KUTTA SLICE.

      IF (I .LT. ILE) GO TO 83
      IF (I .GT. ITE) GO TO 70

C                      AIRFOIL B. C.

      J=JUP
      DIAG(J)=DIAG(J)+CYYC(J)-CYYBUC
      SUP(J)=0.0
      SUB(J)=CYYBUD
      RHS(J)=RHS(J)+CYYD(J)*P(J-1,I)-CYYC(J)*P(J,I)+CYYU(J)*P(J+1,I)
     1          -(-CYYBUC*P(J,I)+CYYBUD*P(J+1,I)+FXUBC(I))
      J=JLOW
      DIAG(J)=DIAG(J)+CYYC(J)-CYYBLC
      SUP(J)=CYYBLD
      SUB(J)=0.0
      RHS(J)=RHS(J)+CYYD(J)*P(J-1,I)-CYYC(J)*P(J,I)+CYYU(J)*P(J+1,I)
     1          -(-CYYBLC*P(J,I)+CYYBLD*P(J-1,I)+FXLBC(I))
      GO TO 83

C                      KUTTA SLICE CHANGE.
   70 CONTINUE
      RHS(JLOW) = RHS(JLOW) + CYYU(JLOW)*PJUMP(I)
      RHS(JUP)  = RHS(JUP)  - CYYD(JUP )*PJUMP(I)
   83 CONTINUE

C                      INSERT WALL B. C.

      IVAL = 1
      CALL BCEND

C                      COMPUTE MAX RESIDUAL.

      IF ( .NOT. OUTERR) GO TO 110
      DO 100  J=JBOT,JTOP
      ARMS = ABS(RHS(J))
      IF (ARMS .GT. BIGRL) GO TO 90
      GO TO 100
   90 CONTINUE
      BIGRL = ARMS
      IRL   = I
      JRL   = J
  100 CONTINUE
  110 CONTINUE
C
C***********   ADD PXT
C
      DO 300 J = JBOT,JTOP
      DIAG(J) = DIAG(J) - EPSX
  300 RHS(J) = RHS(J) - EPSX*(P(J,I-1)-POLD(J,I2))

C                      SOLVE TRIDIAGONAL MATRIX EQUATION.

      DNOM       = 1.0 / DIAG(JBOT)
      SAVE(JBOT)=SUB(JBOT)*DNOM
      RHS(JBOT)  = RHS(JBOT) * DNOM
      DO 120 J=J1,JTOP
      DNOM=1./(DIAG(J)-SUP(J)*SAVE(J-1))
      SAVE(J)=SUB(J)*DNOM
      RHS(J)=(RHS(J)-SUP(J)*RHS(J-1))*DNOM
  120 CONTINUE
      DO 130 K=1,J2
      J       = JTOP - K
      RHS(J) = RHS(J) - SAVE(J) * RHS(J+1)
  130 CONTINUE

C                      COMPUTE NEW P.

      DO 140 J=JBOT,JTOP
      P(J,I) = P(J,I) + RHS(J)
  140 CONTINUE

C                      COMPUTE MAX ERROR

      IF ( .NOT. OUTERR) GO TO 180
      DO 170  J = JBOT,JTOP
      ARMS = ABS(RHS(J))
      IF (ARMS .GT. ERROR) GO TO 160
      GO TO 170
  160 CONTINUE
      ERROR = ARMS
      IERROR = I
      JERROR = J
  170 CONTINUE
  180 CONTINUE
      IF (AK .GT. 0.0) GO TO 195
      IF (I .NE. IDOWN-1) GO TO 195

C                      SET P(IDOWN+1) = P(IDOWN-1) TO OBTAIN
C                      CENTERED VELOCITY AT IDOWN FOR SUPERSONIC
C                      FREESTREAM FLOW.

      DO 190 J=JMIN,JMAX
      P(J,IDOWN+1) = P(J,IDOWN-1)
  190 CONTINUE
  195 CONTINUE
      ISAVE = I2
      I2    = I1
      I1    = ISAVE
      IM2=I-1
  200 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TRAP(X,Y,N,SUM)
C                 INTEGRATE Y DX BY TRAPEZOIDAL RULE
C                 N IS THE NUMBER OF (X,Y) POINTS AND SUM IS THE
C                 RESULTING INTEGRAL.
C                 CALLED BY - CDCOLE, DRAG, PITCH, REDUB.
C                 INTEGRAL
      DIMENSION X(1),Y(1)
      SUM = 0.
      NM1 = N-1
      DO 10 I=1,NM1
      Z = X(I+1) - X(I)
      W = Y(I+1) + Y(I)
      SUM = SUM + Z*W
   10 CONTINUE
      SUM = .5*SUM
      RETURN
      END
```

```
      SUBROUTINE VROOTS
C                 COMPUTE CONSTANTS BETA0,BETA1,BETA2,PSI0,PSI1,PSI2,
C                 USED IN FORMULA FOR VORTEX IN SLOTTED WIND TUNNEL
C                 WITH SUBSONIC FREESTREAM
C                 CALLED BY - FARFLD.

      COMMON /COM12/ F        , H        , HALFPI  , PI       , RTKPOR  ,
     1               TWOPI
      COMMON /COM15/ B        , BETA0    , BETA1   , BETA2    , PSI0    ,
     1               PSI1     , PSI2

      ERROR = .00001
C                 CALCULATE BETA0
      BETA0 = 0.
      DO 10 I = 1,100
      TEMP = BETA0
      Q = -F*TEMP + RTKPOR
      BETA0 = ATAN(Q)
      DBETA = ABS(TEMP - BETA0)
      IF(DBETA .LT. ERROR) GO TO 15
   10 CONTINUE
      N = 0
      GO TO 9999
   15 CONTINUE
C                   CALCULATE BETA1
      BETA1 = 0.
      DO 20 I=1,100
      TEMP = BETA1
      Q = -F*(TEMP + PI) + RTKPOR
      BETA1 = ATAN(Q)
      DBETA  = ABS(BETA1 - TEMP)
      IF(DBETA .LT. ERROR) GO TO 25
   20 CONTINUE
      N = 1
      GO TO 9999
   25 CONTINUE
C                   CALCULATE BETA2
      BETA2 = 0.
      DO 30 I=1,100
      TEMP = BETA2
      Q = -F*(TEMP - PI) + RTKPOR
      BETA2 = ATAN(Q)
      DBETA  = ABS(BETA2 - TEMP)
      IF(DBETA  .LT. ERROR) GO TO 35
   30 CONTINUE
      N = 2
      GO TO 9999
   35 CONTINUE
C                   COMPUTE PSI0,PSI1,PSI2
      TEMP = TAN(BETA0)
      PSI0 = (1. + F/(1. + TEMP*TEMP))
      PSI0 = 1.0 / PSI0
      TEMP = TAN(BETA1)
      PSI1 = (1. + F/(1.+ TEMP*TEMP))
      PSI1 = 1.0 / PSI1
      TEMP = TAN(BETA2)
      PSI2 = (1. + F/(1. + TEMP*TEMP))
      PSI2=1.0/PSI2
      RETURN
C                   ABNORMAL STOP  IF ITERATIONS FOR BETAS DID NOT CONVERGE
 9999 CONTINUE
      WRITE(6,1000) N
 1000 FORMAT(35H1ABNORMAL STOP IN SUBROUTINE VROOTS/
     1  37H0NONCONVERGENCE OF ITERATION FOR BETA,  I1)
      STOP
      END
```

REFERENCES

1. Murman, E. M., Bailey, F. R., and Johnson, M. H.: TSFOIL - A Computer Code for 2-D Transonic Calculations, Including Wind-Tunnel Wall Effects and Wave-Drag Evaluation. NASA SP-347, Part II, 1975, pp. 769-788.

2. Krupp, J. A.: The Numerical Calculation of Plane Steady Transonic Flows Past Thin Lifting Airfoils. Ph.D. Dissertation, University of Washington, 1971.

3. Krupp, J. A. and Murman, E. M.: Computation of Transonic Flows Past Lifting Airfoils and Slender Bodies. AIAA Journal, vol. 10, no. 7, July 1972, pp. 880-886.

4. Bailey, F. R. and Ballhaus, W. F.: Relaxation Methods for Transonic Flow About Wing-Cylinder Combinations and Lifting Swept Wings. Lecture Notes in Physics, vol. 19, Springer-Verlag, 1972, pp. 2-9.

5. Ballhaus, W. F. and Bailey, F. R.: Numerical Calculation of Transonic Flow About Swept Wings. AIAA Paper No. 72-677, June 1972.

6. Murman, E. M.: Analysis of Embedded Shock Waves Calculated by Relaxation Methods. AIAA Journal, vol. 12, no. 5, May 1974, pp. 626-632.

7. Murman, E. M. and Cole, J. D.: Inviscid Drag at Transonic Speeds. AIAA Paper No. 74-540, June 1974.

8. Murman, E. M.: Computation of Wall Effects in Ventilated Transonic Wind Tunnels. AIAA Paper No. 72-1007, Sept. 1972.

9. Murman, E. M.: A Relaxation Method for Calculating Transonic Flows with Detached Bow Shocks. Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics. Lecture Notes in Physics, vol. 19, July 1972, pp. 201-206.

10. Lock, R. C.: Research in the UK on Finite Difference Methods for Computing Steady Transonic Flows. RAE Tech. Memo. Aero. 1644, July 1975.

11. Albone, C. M., Catherall, D., Hall, M. G., and Joyce, G.: An Improved Numerical Method for Solving the Transonic Small-Perturbation Equation for the Flow Past a Lifting Airfoil, RAE Tech. Rept. 74056, August 1974.

12. Jameson, A.: Transonic Flow Calculation, Lecture Series 87 von Karman Institute for Fluid Dynamics, Computational Fluid Dynamics, March 15-19, 1976.

13. Nixon, D.: Calculation of Transonic Flows Using an Extended Integral Equation Method. AIAA Jour., vol. 15, No. 3, March 1977, pp. 295-296.

14. Olsen, J.: Subsonic and Transonic Flow Over Sharp and Round Nosed Nonlifting Airfoils. Ohio State Univ. Ph.D. Thesis, 1976.

15. Kacprzynski, J. J., Ohman, L., Garabedian, P., and Korn, D.: Analysis of the Flow Past a Shockless Lifting Airfoil in Design and Off Design Conditions. Aeronautical Report LR-554, National Research Council of Canada, Nov. 1971.

16. Baldwin, B., Turner, J., and Knechtel, E.: Wall Interference in Wind Tunnels with Slotted and Porous Boundaries at Subsonic Speeds. NACA TN 3176, 1954.

17. Collins, D. J. and Krupp, J. A.: Experimental and Theoretical Investigations in Two-Dimensional Transonic Flow. AIAA Journal, vol. 12, no. 6, June 1974, pp. 771-778.

18. Bailey, F. R. and Ballhaus, W. F.: Comparisons or Computed and Experimental Pressures for Transonic Flows About Isolated Wings and Wing-Fuselage Configurations. NASA SP-347, Part I, 1975, pp. 1213-1232.

19. Murman, E. M.: Analysis of Embedded Shock Waves Calculated by Relaxation Methods. Proceedings of AIAA Computational Fluid Dynamics Conference, Palm Springs, CA, July 1973, pp. 27-40.

20. Murman, E. M. and Cole, J. D.: Calculation of Plane Steady Transonic Flows. AIAA Journal, vol. 9, no. 1, Jan. 1971, pp. 114-121.

21. Bailey, F. R.: On the Computation of Two- and Three-Dimensional Steady Transonic Flows by Relaxation Methods. VKI Lecture Series "Progress in Numerical Fluid Dynamics," Von Karman Institute for Fluid Dynamics, Rhode-St-Genese, Belgium, Feb. 11-15, 1974.

22. Lock, R. C.: Test Cases for Numerical Methods in Two-Dimensional Transonic Flow. AGARD Rept. No. 575, Nov. 1970.

23. Steger, J. L. and Lomax, H.: Transonic Flow About Two-Dimensional Airfoils by Relaxation Procedures. AIAA Journal, vol. 10, no. 1, 1972, pp. 49-54.

24. Garabedian, P. R. and Korn, D. G.: Analysis of Transonic Airfoils. Comm. Pure Applied Math., vol. XXIV, 1971, pp. 841-851.

25. Sells, C. C. L.: Plane Subcritical Flow Past a Lifting Airfoil. RAE TR 67146, June 1967.

26. Jameson, A.: Iterative Solution of Transonic Flows Over Airfoils and Wings, Including Flows at Mach 1. Comm, Pure Applied Math., vol. 27, 1974, pp. 283-309.

27. Yoshihara, H.: A Survey of Computational Methods for 2D and 3D Transonic Flows with Shocks. Advances in Fluid Dynamics, AGARD Lecture Series No. 64, 1973, pp. 6-1 to 6-35.

28. Magnus, R. M.: The Direct Comparison of the Relaxation Method and the Pseudo-Unsteady Finite Difference Method for Calculating Steady Planar Transonic Flow. TN-73-SP03, General Dynamics-Convair Aerospace Division, 1973.

29. Melnik, R. E. and Ives, D. C.: On Viscous and Wind-Tunnel Wall Effects in Transonic Flows Over Airfoils. AIAA Paper No. 73-660, July 1973.

30. Osborne, J.: A Selection of Measured Transonic Flow Pressure Distributions for the NACA 0012 Airfoil: Provisional Data from an NPL Tunnel. Aero. Dept., RAE, June 1971.

31. Hafez, M. M. and Cheng, H. K.: Convergence Acceleration and Shock Fitting for Transonic Aerodynamic Computations. AIAA Paper No. 75-51, Jan. 1975.

32. Lomax, H. and Martin, E. D.: Fast Direct Numerical Solution of the Nonhomogeneous Cauchy-Riemann Equations. Journal Comp. Phys., vol. 15, 1974, pp. 55-80.

33. Martin, E. D.: A Fast Semidirect Method for Computing Transonic Aerodynamic Flows. AIAA 2nd Computational Fluid Dynamics Conference (Proceedings), Hartford, CT, June 19-20, 1975, pp. 162-174.

34. Hafez, M. M. and Cheng, H. K.: Shock-Fitting Applied to Relaxation Solutions of Transonic Small-Disturbance Equations. AIAA Jour. vol. 15, no. 6, June 1977, pp. 786-793.

1. Report No. NASA CR-3064	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Operational Manual for Two-Dimensional Transonic Code TSFOIL		5. Report Date December 1978
		6. Performing Organization Code
7. Author(s) Stephen S. Stahara		8. Performing Organization Report No. NEAR TR 94
9. Performing Organization Name and Address Nielsen Engineering & Research, Inc. 510 Clyde Avenue Mountain View, California 94043		10. Work Unit No.
		11. Contract or Grant No. NAS2-8648
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D. C. 20546		13. Type of Report and Period Covered Contractor Report
		14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

A detailed operating manual is presented for the computer code TSFOIL recently developed by Murman, Bailey, and Johnson at NASA/Ames Research Center. This code solves the two-dimensional, transonic, small-disturbance equations for flow past lifting airfoils in both free air and various wind-tunnel environments by using a variant of the finite-difference method initially proposed by Murman and Cole. A description of the theoretical and numerical basis of the code is provided, together with complete operating instructions and sample cases for the general user. In addition, a programmer's manual is also presented to assist the user interested in modifying the code. Included in the programmer's manual are a dictionary of subroutine variables in common and a detailed description of each subroutine.

17. Key Words (Suggested by Author(s)) Transonic Theory Steady Flow Airfoils Wind Tunnel Flows Finite-Difference Relaxation	18. Distribution Statement UNCLASSIFIED-UNLIMITED Cat. 02		
19. Security Classif. (of this report) UNCLASSIFIED	20. Security Classif. (of this page) UNCLASSIFIED	21. No. of Pages 179	22. Price* $9.00

*For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA-Langley, 1978

END

MAR 2 1979